

SENG2011 – Workshop on Reasoning about Programs: from Specification to Implementation

Created: 22 Mar 2017

Proposal Last Updated: 29 Mar 2017

Offering Details:

Key Details and Contacts

Key Course Details

Course Name (Official)	Workshop on Reasoning about Programs: from Specification to Implementation
Standard Name (SIMS)	Program Reasoning Workshop
Course Code	SENG2011
Units of Credit (UOC)	6
Career	Undergraduate
Level	2
First semester and year the revised changes will take effect	2018 Semester 2

Contact Details

Proposal Proponent	Name	Email	Role
	Fethi Rabhi	f.rabhi@unsw.edu.au	Professor, School of Computer Science and Engineering
Proposal Author(s)	Name	Email	Role
	Gabriele Keller	keller@cse.unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering
	John Shepherd	jas@cse.unsw.edu.au	Deputy Head of School (Education), School of Computer Science and Engineering
	Kai Engelhardt	kaie@cse.unsw.edu.au	–
Proposal Contact	Name	Email	Role
	Fethi Rabhi	f.rabhi@unsw.edu.au	Professor, School of Computer Science and Engineering
Optional Additional Endorsers	Not specified		
Academic Unit responsible for course	School of Computer Science and Engineering		
Parent Academic Unit	Faculty of Engineering		

Proposal Concept

Summary of Proposal

Summary of Proposal	<p>This proposal is a revision of an existing course SENG2011 Software Engineering Workshop 2A. The main changes are</p> <ul style="list-style-type: none"> • a new sequential order with respect to COMP2111 • removal of a dependence on SENG1031, which is being eliminated, • transfer of some more advanced material, on data refinement, from COMP2111 to SENG2011 • elimination of Project Management from COMP2011 <p>See the attachment for proposed content.</p>
---------------------	--

Justification for proposal

Justification for Proposal	<p>The proposed changes were developed as a part of the Software Engineering Program review.</p> <p>The course, as it was being taught, and in relation to others, suffered from a number of weaknesses:</p> <ul style="list-style-type: none"> • COMP2111 should be a prerequisite not co-requisite • SENG1031 ceased providing the inputs previously expected by this course • The primary material in this course requires development of individual skills, making its combination with Project Management a poor fit. <p>The attached documentation expands on the history of these courses and considerations concerning their fit in CSE degrees.</p>
-----------------------------------	---

Attachments

Attach documentation to this proposal	No.	Description	File(s)
	1	Background to Revision of COMP2111 and SENG2011	background.pdf
	2	Proposal for new SENG2011	newSENG2011.pdf

Learning and Teaching

Learning & Teaching development and support

Are there Learning & Teaching space requirements for the course beyond those that can be accommodated by CATS spaces?	No
Have you discussed with the Learning Centre and Learning and Teaching what language and/or academic skills development resources and/or which teaching and learning strategies might be suited to this course?	No
Are many students in this course at a key transition point where their academic skills are likely to need development, e.g. from one kind of educational institution or type of program to another or into education after a significant break?	No

Consultation

Internal consultation

Internal Consultation	Consultants	None specified
	Details	This proposal is as a result of a review of the Software Engineering degree which was conducted between July and October. A steering group comprising academics from the School as well as external people was formed and met on a regular basis. A subpanel met to consider issues concerning COMP2111 and SENG2011.
	Attachments	None specified

External consultation

External Consultation	Consultants	None specified
	Details	None specified
	Attachments	None specified

Interested Parties	Not specified
---------------------------	---------------

Related Proposals

Related Proposals	Code	Proposal Name	Type	Date	Status
	COMP2111	System Modelling and Design	Course Revision (UG)	Mar 2017	Draft Proposal
	SENG2021	Requirements and Design Workshop	Course Revision (UG)	Mar 2017	Submitted

Endorsements and Comments

Endorsement history	No endorsements have been recorded for this proposal (yet).
----------------------------	---

CommentsNo comments posted

Administration:

Key Course Details

Key Admin Details

Course Name (Official)	Workshop on Reasoning about Programs: from Specification to Implementation
Student System ID	00063058
Can course be taken as General Education elective?	No
Field of Education	020103 – Programming

Course Review

Next course review date	January 01, 2020
--------------------------------	------------------

Provide details of any particular factors that need to be considered at that review.

Summary of Evaluations

The course has had a significant number of changes of lecturer in charge: 2013-14: Carroll Morgan, 2015: Ron van der Meyden, 2016-17: Albert Nymeyer. Each has taken a different approach to the content, in particular with respect to which verification tool was used (B-tool or Dafny), and the degree of reliance placed on that tool. Teaching the course has been a learning exercise for all involved with respect to both the capabilities of the tools used and the extent to which they are accessible to students, and has resulted in significant changes each year.

One of the main problems with the course, borne out in evaluations, is a lack of coherence, from combining project management (necessitating a group-work project) and with a focus on specification and the development of correct programs from specifications (implying a focus on development of individual skills). This is being addressed in the present revision by removing the project management component.

Another issue is that that students find the material challenging, with one of the main reasons being that prerequisite theoretical knowledge from COMP2111 was being taught as a co-requisite, leaving insufficient time for assimilation before intensive use. This is being addressed in the present revision by ordering COMP2111 and SENG2011 in successive semesters.

The main issue for future review is then the extent to which the improved coherence, extra time on the material resulting from the changes proposed, and increased lecturer-in-charge experience with the course and tools used (assuming stable staffing) helps to improve satisfaction ratings, learning outcomes and accessibility of the content to students.

2016 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%**

Albert Nymeyer's 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

Delivery and Attendance				
Campus administering the Course	Sydney			
Teaching Shares by School/Faculty	School	Teaching Share (%)		
	School of Computer Science and Engineering	100		
	Total Share	100		
Semesters the course is offered		Summer Semester	Semester 1	Semester 2
	2017	No	No	No
	2018	No	No	Yes
	2019	No	No	Yes
	2020	No	No	Yes
Teaching mode and contact hours	Standard Offering Mode			
Standard offering contact hours per week	Learning Activity	Hours/Week		
	Lecture	3		
	Tutorial/Laboratory	0.5		
	Tutorial	0		
	Laboratory	0		
	Web-based Online Learning Activity	0		
	Clinical/Fieldwork	0		
	Distance Learning	0		
	Seminar	0		
	Studio	0		
	Meeting/Consultation	0		
	Total Hours per week	3.5		
Primary delivery mode	Classroom			
Secondary delivery modes	Online, Directed Research			
Additional information about the delivery modes for this course	<p>In following the workshop nature of the course, lectures are mixed-mode, sometimes using standard lecture format for presentation of new content, but often interlacing hands-on group and individual exercises.</p> <p>Rather than 1 hour tutorials with a large number (average 20) of students, students are formed into smaller learning groups (8 students in past iterations) and given more individually focussed attention from a "mentor". In past iterations, these learning groups served as project groups for a major project. In the present revision, the major project component is dropped for a focus on individual skill development.</p>			

Staff			
Staff associated with course			
Course Convenor	Not specified		
Administrative Contact	Name	Email	Role
	Cassandra Nock	cassandra@CSE.UNSW.EDU.AU	Administration Manager, School of Computer Science and Engineering

Supplementary Information:

Resources

Student Resources

Prescribed Resources	1.	System Modelling and Design	Other
	Resource Type	Text and teaching material for the prerequisite course COMP2111	
	Additional Details	The rigorous techniques and strategies are taught in the course COMP2111	
Recommended Resources	None specified		

Experience and Assumed Knowledge

Industrial Experience Component

Industrial Experience Component	None
--	------

Assumed Knowledge

Assumed Knowledge	Learning outcomes from COMP2111.
--------------------------	----------------------------------

Academic Structure:

Academic Structure

Prerequisites

Prerequisite courses	COMP2111 - System Modelling and Design (UG)
Prerequisite programs	Not specified
Prerequisite streams	Not specified
Prerequisite conditions	Not specified

Exclusions

Excluded Courses	Not specified
Excluded Programs	Not specified
Excluded Streams	Not specified

Equivalent

Equivalent courses	Not specified
---------------------------	---------------

Assessment

Assessment

Grading Basis	Standard UNSW grades (e.g. HD, DN, CR, PS, FL)
----------------------	--

Assessment items and their relationship to Course Learning Outcomes

Assessment Title	Assessment Type	Weight (%)
1 Input-output interface specifications	Assignment	10%
Assessment Description:	Use of predicate logic and quantification for input-output specifications of list-based routines. Feedback : marked assignment with comments.	
2 Intermediate assertions and invariants	Assignment	20%
Assessment Description:	Correctness proof for a simple program involving looping over a list, using invariants, intermediate assertions and simple numeric variant. Feedback: marked assignment with comments.	
3 Refinement	Assignment	20%
Assessment Description:	Development of a correct program by a refinement process. Feedback: marked assignment with comments.	
4 Data refinement	Assignment	20%
Assessment Description:	Development of an efficient datatype implementation from an inefficient specification using coupling invariants. Feedback: marked assignment with comments.	
5 Moderate scale example	Assignment	30%
Assessment Description:	Development of a correct moderately complex program through a combination of the above techniques, and composed of multiple modules. Done as a group with subgroups working on subcomponents, to illustrate development by (formal) contract. Feedback: marked assignment with comments.	
Total Weight		100%

Input-output interface specifications

- Develop a rigorous specification from a set of requirements for a system, particularly a software system.
- Be able to use a tool supporting formal specification and verification.

Intermediate assertions and invariants

- Develop a rigorous specification from a set of requirements for a system, particularly a software system.
- Apply rigorous patterns of reasoning for establishing correctness of a program.
- Apply rigorous patterns of reasoning for establishing termination of a program.
- Be able to use a tool supporting formal specification and verification.

Refinement

- Develop a rigorous specification from a set of requirements for a system, particularly a software system.
- Apply rigorous patterns of reasoning for establishing correctness of a program.
- Apply rigorous patterns of reasoning for establishing termination of a program.
- Apply patterns of reasoning whereby a correct program implementing a specification can be developed through a process of refinement.
- Be able to use a tool supporting formal specification and verification.

Data refinement

- Develop a rigorous specification from a set of requirements for a system, particularly a software system.
- Apply rigorous patterns of reasoning for establishing correctness of a program.
- Apply rigorous patterns of reasoning for establishing termination of a program.
- Apply patterns of reasoning whereby an inefficient data type specification is replaced by a more efficient implementation.
- Be able to use a tool supporting formal specification and verification.

Moderate scale example

- Be able to use a tool supporting formal specification and verification.
- Apply patterns of reasoning whereby a correct program implementing a specification can be developed through a process of refinement.
- Apply rigorous patterns of reasoning for establishing termination of a program.
- Apply rigorous patterns of reasoning for establishing correctness of a program.
- Develop a rigorous specification from a set of requirements for a system, particularly a software system.
- Apply patterns of reasoning whereby an inefficient data type specification is replaced by a more efficient implementation.

Curriculum Mapping

Course Learning Outcomes

Specify the learning outcomes that students should achieve upon successful completion of this course

- 1 Develop a rigorous specification from a set of requirements for a system, particularly a software system.
- 2 Apply rigorous patterns of reasoning for establishing correctness of a program.
- 3 Apply rigorous patterns of reasoning for establishing termination of a program.
- 4 Apply patterns of reasoning whereby a correct program implementing a specification can be developed through a process of refinement.
- 5 Apply patterns of reasoning whereby an inefficient data type specification is replaced by a more efficient implementation.
- 6 Be able to use a tool supporting formal specification and verification.

Teaching strategies and Rationale

Teaching Strategies and Rationale

The course uses lectures, individual work, team work and mentoring to develop skills and experience in the process of producing a specification of a system given the requirements of a proposed system, and then using that specification through a systematic refinement process to produce a correct implementation of the specification.

Course Aims

Course Aims

This course is aimed at developing the ability of students to build a rigorous specification of a system from a set of requirements, and to develop an implementation that correctly implements that specification.

Publications and Marketing:

Publications

Course Description

Description of course that can be used in online publications (e.g. Handbook website, Faculty websites or other online catalogue systems)

This is a workshop course aimed at developing the skills of writing precise specifications of programs and translating these specifications into correct implementations. The course applies the rigorous modelling and verification techniques introduced in COMP2111 to a diverse and increasingly complex set of problems. Further methods for reasoning about programs are introduced, including methods for reasoning about termination, program refinement and data refinement. The primary learning outcome is to develop students' abilities to apply these ideas to structure their thinking about programs, but the course may use a formal verification tool to support learning.

Key Search Terms

List key search terms that might be used to search for this course (e.g. via the Handbook or Google searches).

reasoning about programs
formal specification
verification
program refinement
data refinement
verification tools