

COMP1531 – Software Engineering Fundamentals

Created: 23 Apr 2015

Proposal Last Updated: 29 Apr 2016

Offering Details:

Key Details and Contacts

Key Course Details

Course Name (Official)	Software Engineering Fundamentals
Standard Name (SIMS)	Software Eng Fundamentals
Course Code	COMP1531
Units of Credit (UOC)	6
Career	Undergraduate
Level	1
First Semester and Year the Course will be offered	2017 Semester 1
Does this new course replace another existing course?	Yes
Courses to be replaced	SENG1031 - Software Engineering Workshop 1 (UG)

Contact Details

Proposal Proponent	Name	Email	Role
	John Shepherd	jas@cse.unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering
Proposal Author(s)	Not specified		
Proposal Contact	Name	Email	Role
	John Shepherd	jas@cse.unsw.edu.au	Senior Lecturer, 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 course provides an introduction to software engineering principles and practice, including software lifecycle, methodologies, teamwork and web-based architectures.</p> <p>The course is part of CSE's core syllabus redevelopment. It is intended to be the first course in software engineering taken by all students enrolled in a CSE-run degree or any dual-award program that includes a CSE-run program. It effectively replaces SENG1031, and opens up study of software engineering to all CSE students.</p> <p>This course is not intended to be exclusive to CSE students. Students from other disciplines are welcome to enrol.</p>
---------------------	---

Justification for proposal

Justification for Proposal	<p>As part of on-going curriculum development, CSE has redesigned its lower-level courses based on the ACM Computing Curriculum, to include all of the topics that we believe every CSE graduate should know about when they graduate. This has led to the development of five new courses, which effectively replace the existing COMP1917, COMP1927 and COMP2911 courses. In addition, SENG1031 (the first software engineering workshop) will be replaced by an introductory software engineering course that all students in CSE degrees must take. Also, the other level-2 courses COMP2121 and COMP2041 will be re-designed to complement the new core syllabus.</p> <p>Note that while these courses are targeted at students in CSE degrees, we encourage students from other programs to enrol if they want a more comprehensive introduction to computing than what is available in the service courses COMP1911, COMP1921 and ENGG1811. These service courses will, however, be retained for the majority of Engineering students who do not think they need such a detailed view of computing.</p>
----------------------------	--

Anticipated Enrolments

Anticipated Enrolments for next 3 years	2017	2018	2019
	400	400	400

Attachments	
Attach documentation to this proposal	None attached

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	Jingling Xue (Professor, School of Computer Science and Engineering)
	Details	Jingling Xue ran the Working Group which developed the new core syllabus. The Working Group included academics across a range of computing disciplines and levels.
	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
	COMP1511	Introduction to Programming	New Course (UG)	Apr 2015	Submitted
	COMP1521	Computer Systems Fundamentals	New Course (UG)	Apr 2015	Draft Proposal

Endorsements and Comments

Endorsement history	No endorsements have been recorded for this proposal (yet).
Comments	No comments posted

Administration:

Key Course Details

Key Admin Details

Course Name (Official)	Software Engineering Fundamentals
Student System ID	A Student System ID will be generated once this course is approved.
Can course be taken as General Education elective?	Yes
Field of Education	020103 – Programming

Course Review

Next course review date	December 01, 2018
Provide details of any particular factors that need to be considered at that review.	The course will be reviewed after each offering in 2017 to check how effectively it is meeting its objectives. A formal review will be conducted at the end of 2018.

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
	2015	No	No	No
	2016	No	Yes	Yes
	2017	No	Yes	Yes
	2018	No	Yes	Yes
	2019	No	Yes	Yes

Teaching mode and contact hours	Standard Offering Mode
--	------------------------

Standard offering contact hours per week	Learning Activity	Hours/Week
	Lecture	3
	Tutorial/Laboratory	3
	Tutorial	1
	Laboratory	2
	Web-based Online Learning Activity	0
	Clinical/Fieldwork	0
	Distance Learning	0
	Seminar	0
	Studio	0
	Meeting/Consultation	1
Total Hours per week	10	

Primary delivery mode	Classroom
------------------------------	-----------

Secondary delivery modes	Online
---------------------------------	--------

Additional information about the delivery modes for this course	All course materials will be available online. Students must attend tutorial/laboratories. Students should attend lectures. Fortnightly meetings with project mentors are compulsory.
--	---

Staff

Staff associated with course

Course Convenor	Name	Email	Role
	Boualem Benatallah	boualem@cse.unsw.edu.au	Professor, School of Computer Science and Engineering
	John Shepherd	jas@cse.unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering
Administrative Contact	Not specified		

Supplementary Information:

Resources

Student Resources

Prescribed Resources	None specified
-----------------------------	----------------

Recommended Resources	None specified
------------------------------	----------------

Experience and Assumed Knowledge

Industrial Experience Component

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

Assumed Knowledge

Assumed Knowledge	We assume no prior programming experience. Students should have Mathematics background equivalent to NSW HSC Mathematics and have a good command of English (IELTS 7.5 or better).
--------------------------	--

Academic Structure:

Academic Structure

Prerequisites

Prerequisite courses	Not specified
Prerequisite programs	Not specified
Prerequisite streams	Not specified
Prerequisite conditions	Not specified

Exclusions

Excluded Courses	SENG1031 - Software Engineering Workshop 1 (UG)
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 Final Exam	Examination	50%
Assessment Description:	The final examination will include both Practical and Written components. Feedback via final mark.	
2 Project	Project	50%
Assessment Description:	Over the course of the semester, students will work in teams to specify, design and implement a non-trivial software system. Fortnightly meetings with a team mentor will provide feedback on progress. Peer assessment and individual reflective journals will allow students to provide feedback on their own and each other's work.	
Total Weight		100%
Final Exam		
Project		

Curriculum Mapping

Course Learning Outcomes

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

- describe the phases of software development and life-cycle of software - and illustrate them from experience
- effectively choose and use a range of project management and software development tools
- describe common behaviour that contribute to the effective functioning of a team and identify necessary roles in a software development team
- articulate software design principles and use a design paradigm to design a simple software system (e.g., simple Web application in MVC)
- demonstrate robust coding practices (e.g., handling exceptions, following coding standards)
- describe effective coding validation and verification techniques (e.g., code reviewing, fault logging, a range of test measures)
- demonstrate effective usage of testing fundamentals (e.g., unit tests, integration tests, test plan/cases, test automation)

Teaching strategies and Rationale

Teaching Strategies and Rationale	Lectures will provide a summary of the material, but the primary learning environment will be the group project. Tutorials will allow students to discuss principles and practice design. Lab classes will allow them to practice with relevant tools. Group meetings will allow students to discuss design and implementation strategies.
Course Aims	
Course Aims	This course aims to expose students to: <ul style="list-style-type: none">● basic elements of software engineering - derived from the lifecycle of a software system, including requirements elicitation, analysis and specification; design; construction; verification and validation; deployment; and operation and maintenance● software engineering methodologies, processes, measurements, tools and techniques● Web-based system architecture and development practices on Web platforms.

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)

Introduction to software engineering principles: exposing students to basic software lifecycle concepts, modern development methodologies, conceptual modeling and how these activities relate to programming. Introducing the basic notions of team-based project management via conducting a small project involving a simple application.

Key Search Terms

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

programming
computing
software engineering