

COMP2511 – Object-oriented Design

Created: 29 Apr 2016

Proposal Last Updated: 29 Apr 2016

Offering Details:

Key Details and Contacts

Key Course Details

Course Name (Official)	Object-oriented Design
Standard Name (SIMS)	Object-oriented Design
Course Code	COMP2511
Units of Credit (UOC)	6
Career	Undergraduate
Level	2
First Semester and Year the Course will be offered	2018 Semester 1
Does this new course replace another existing course?	Yes
Courses to be replaced	Not specified

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 the principles and practice of object-oriented design of software systems, including OO modelling and OO-programming.</p> <p>The course is part of CSE's core syllabus redevelopment. It is intended to be the first course in object-oriented design, taken by all students enrolled in a CSE-run degree or any dual-award program that includes a CSE-run program. It effectively replaces COMP2511, and opens up study of software engineering to all CSE students.</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	2018	2019	2020
	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
	COMP1531	Software Engineering 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)	Object-oriented Design
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, 2019
Provide details of any particular factors that need to be considered at that review.	The course will be reviewed after each offering in 2018 to check how effectively it is meeting its objectives. A formal review will be conducted at the end of 2019.

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
	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	0
	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	7

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.
---	---

Staff

Staff associated with course

Course Convenor	Name	Email	Role
	John Shepherd	jas@cse.unsw.edu.au	Senior Lecturer, School of Computer Science and Engineering
	Wayne Wobcke	wobcke@cse.unsw.edu.au	Associate Professor, 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 that students have experience programming in a procedural language.

Academic Structure:

Academic Structure

Prerequisites

Prerequisite courses	COMP1531 - Software Engineering Fundamentals (UG)
Prerequisite programs	Not specified
Prerequisite streams	Not specified
Prerequisite conditions	Not specified

Exclusions

Excluded Courses	COMP2911 - Engineering Design in Computing (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 OO Implementation	Assignment	15%
Assessment Description:	Students will implement an OO design and develop a testing strategy to demonstrate the quality of their implementation. Tutors will provide feedback on the code.	
2 OO Design	Assignment	15%
Assessment Description:	Students will develop an OO design based on a real-world problem specification. Tutors will assess the designs and provide feedback to students. Designs will also be presented and discussed in tutorial classes.	
3 Final Exam	Examination	70%
Assessment Description:	The final examination will include both Practical and Written components. Feedback via final mark.	
Total Weight		100%

OO Implementation

OO Design

Final Exam

Curriculum Mapping

Course Learning Outcomes

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

- 1 design appropriate solutions to medium-scale problems using Java
- 2 apply the object-oriented design principles such as separation of concerns, responsibility analysis, and design by contract
- 3 apply object-oriented analysis and design practice to complex software systems
- 4 create and refactor medium-scale object-oriented programs in Java using appropriate design principles
- 5 describe and use the most important OO design patterns

Teaching strategies and Rationale

Teaching Strategies and Rationale	Lectures will provide a summary of the material, but the primary learning environment will be the labs and assignments. Tutorials will allow students to discuss principles, analyse designs, and practice design. Lab classes will allow them to practice with relevant tools.
Course Aims	
Course Aims	This course aims to: <ul style="list-style-type: none">• provide an introduction of the fundamental principles of object-oriented design• provide an introduction to object-oriented programming and object-oriented design in Java• help students develop sound programming and design skills, problem solving and modeling of real world problems from science, engineering, and economics using the object-oriented paradigm

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 object-oriented design. Object-oriented design patterns. Design by contract. Object-oriented programming in Java.

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
object orientation