UNSW NEW COURSE PROPOSAL

COMP1511 – Introduction to Programming

Created: 22 Apr 2015 Proposal Last Updated: 28 Apr 2016

Offering Details:

Key Details and Contacts

Key Course Details	
Course Name (Official)	Introduction to Programming
Standard Name (SIMS)	Introduction to Programming
Course Code	COMP1511
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	COMP1917 - Computing 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)

Name	Email	Role
Cassandra Nock	cassandra@CSE.UNSW.EDU.AU	Administration Manager, School of Computer Science and Engineering
Kerstin Vintila	z3028233@unsw.edu.au	Administrative Officer, School of Computer Science and Engineering

Proposal Contact

Name	Email	Role
Cassandra Nock	cassandra@CSE.UNSW.EDU.AU	Administration Manager, School of Computer Science and Engineering
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

This course gives an introduction to the theory and practice of programming. It provides an essential introduction the field of computing.

The course is part of CSE's core syllabus redevelopment. It is intended to be the first course taken by all students enrolled in a CSE-run degree or any dual-award program that includes a CSE-run program. It effectively replaces COMP1917.

This course is not intended to be exclusive to CSE students. Students from other disciplines are welcome to enrol, especially if they are considering studying more advanced COMP courses.

Justification for proposal

Justification for Proposal	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, which form the current core of all CSE streams. 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. 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 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.

Anticipated Enrolments

	2017	2018	2019
Anticipated Enrolments for next 3 years	700	700	0

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?	Yes
Details of the key transition point where their academic skills are likely to need development	This is a first computing course and will typically be taken in a student's first semester of study at UNSW (and this is certainly true for students studying CSE programs). The students will typically be making the transition from High School to University. This is no different to the current COMP1917 course, and we will retain the same practices that are used in COMP1917 to assist this transition.

Consultation

Internal consultation

Internal Consultation	**Consultants**	None specified
	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. Other Engineering schools who use our core courses (primarily EET and Mechatronics students) have been consulted and are happy with the proposal.
	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**
	COMP1521	Computer Systems Fundamentals	New Course (UG)	Apr 2015	Draft Proposal
	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)	Introduction to Programming
Student System ID	N/A
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	4
Tutorial/Laboratory	3
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	**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
Richard Buckland	z9700463@unsw.edu.au	Associate Professor, School of Computer Science and Engineering

Administrative Contact	Name	Email	Role
	Kerstin Vintila	z3028233@unsw.edu.au	Administrative Officer, School of Computer Science and Engineering

Supplementary Information:

Resources

Student Resources			
Prescribed Resources	**1.**	**COMP1511 Website**	**Website**
	URL	http://www.cse.unsw.edu.au/~cs1511/	
	Publisher	CSE	
	Additional Details	includes materials, activities, messaging	
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	None

Exclusions	
Excluded Courses	COMP1917 - Computing 1 (UG)
Excluded Programs	Not specified
Excluded Streams	Not specified

Equivalent	
Equivalent courses	COMP1917 - Computing 1 (UG)

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	75%
	Assessment Description:	The final examination will include both Practical and Written components. Feedback via final mark.	
2	Portfolio/Resume	Report	10%
	Assessment Description:	Students will submit a portfolio describing their progress through the course, with reflections on what they learned, what problems they encountered, how they overcame problems, etc. There will be a mid-semester submission for parts of the portfolio, with feedback from tutors on how to improve for the final submission. Additionally, students may submit drafts of the final portfolio to their tutor for comment before making the final submission.	
3	Project	Project	15%
	Assessment Description:	In the second half of the semester, students will complete a major programming project in teams. Students can obtain feedback from tutors on their programs during the development of their solution. Feedback will also be provided in the tutor's commentary on the final submission.	
Total Weight			**100%**

Final Exam

- test and debug programs
- design complete software solutions for simple problems
- distinguish between well-written programs and poorly written programs
- create and use simple data structures
- have a solid understanding of values, storage and addressing
- understand and appropriately use abstraction

Portfolio/Resume

- work in a team to develop software
- distinguish between well-written programs and poorly written programs
- write programs using good programming style
- understand and appropriately use abstraction

Project

- design software solutions for larger problems using abstraction and interfaces
- work in a team to develop software
- test and debug programs
- understand and appropriately use abstraction

Curriculum Mapping

Course Learning Outcomes

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

1	design complete software solutions for simple problems
2	design software solutions for larger problems using abstraction and interfaces
3	distinguish between well-written programs and poorly written programs
4	write programs using good programming style
5	understand and appropriately use abstraction
6	have a solid understanding of values, storage and addressing
7	create and use simple data structures
8	test and debug programs
9	work in a team to develop software

Teaching strategies and Rationale

Teaching Strategies and Rationale	Computing is best learned by practice, and labs and programming assignments are a critical component of the course. These forums allow students to practice design and implementation skills, and to develop teamwork skills. The portfolio will assist in developing students ability to reflect on their own work. Tutorials will provide a forum for students to develop design skills and to practice presentations. Lectures will be split between discussion of concepts, discussion of practical work (and practical demonstrations), revision work, and extension lectures. Study material will be available in both video and note form before each lecture.

Course Aims	
Course Aims	COMP1511 aims to introduce students to the principles and practice of programming, to develop their teamwork and reflective skills, and to prepare a foundation for the future study of computing.

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)	An introduction to problem-solving via software, which assumes no prior programming background. Topics: simple algorithms, simple data structures, code quality, teamwork. The course includes extensive practical work in labs and programming projects.

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