UNSW COURSE REVISION PROPOSAL

ENGG1811 – Computing for Engineers

Created: 02 May 2017 Proposal Last Updated: 04 May 2017

Offering Details:

Key Details and Contacts

Key Course Details	
Course Name (Official)	Computing for Engineers
Standard Name (SIMS)	Computing for Engineers
Course Code	ENGG1811
Units of Credit (UOC)	6
Career	Undergraduate
Level	1
First semester and year the revised changes will take effect	2018 Semester 1

Contact Details			
Proposal Proponent	**Name**	**Email**	**Role**
	Chun Tung Chou	ctchou@cse.unsw.edu.au	Associate Professor, School of Computer Science and Engineering
Proposal Author(s)	**Name**	**Email**	**Role**
	Chun Tung Chou	ctchou@cse.unsw.edu.au	Associate Professor, School of Computer Science and Engineering
Proposal Contact	**Name**	**Email**	**Role**
	Chun Tung Chou	ctchou@cse.unsw.edu.au	Associate 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

ENGG1811 currently consists of the 4 components: (i) 3 weeks on spreadsheets based on OpenOffice Calc; (ii) 3 weeks on OpenOffice Basic; (ii) 4-5 weeks of Matlab; and, (iv) 1 week on how the computing technology is going.

The course certainly has a lot of breath and exposes the students to two very different programming environments. However, there are overheads because students need time to get used to each programming environment. The downside is that there is not enough time to get into deeper topics.

The proposal is that ENGG1811 should focus predominantly on one programming environment so that they can master it well. At the same time, we do not see ENGG1811 as a course which is just about programming. We see that ENGG1811 should cover these components:

1. Programming language
2. Problem solving (computational thinking) and basic computing principles
3. Engineering applications of computing and in particular data processing

The proposed programming language to be used is Python.

The course will also include a minor component on spreadsheets and one on Matlab. These minor components may be in the form of online modules that the students can complete. These two minor components will be designed so that the students can use their knowledge in programming language and basic computing principles (which they learn in the first part of the course) to better understand how to better make use of these computational package. This is to ensure that different components are integrated and connected.

The Python component will take up 10 weeks, while each of the two minor components will take up 1 week.

Justification for proposal

Justification for Proposal

By focusing on one language and a good one, students are given the time to learn it well and become comfortable with it. This will make them more likely to use programming for their future study and work. Moreover, the time freed up by focusing on one language will allow us to introduce more advanced programming skills that engineers need for their work (e.g. data analysis, data cleaning, reading data from files for processing etc.) and some software development skills (e.g. debugging, program testing.)

There were two candidates for the choice of programming language for ENGG1811. They are Matlab and Python. The proposal is use Python for the following reasons:

- Python is a true programming language while Matlab is not.
- Some universities that have a compulsory programming course for engineering students have switched to Python, e.g. Cambridge.
- Python is considered to be a good first language to learn.
- Python is widely used for data analysis which is an important task for engineers.
- Python is free so students can introduce it to their workplace when they go out to work.

The fact that Python is a true programming language is an important one. This will allow the students to learn other programming languages in the future because they will have seen all the important features of a proper programming language.

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 course is designed for first year undergraduate engineering students. These students have studied mathematics in HSC and have the ability to solve mathematical problems analytically. They are also competent in using calculators to solve numerical problems. However, most of the modern day engineering problems can neither be solved analytically nor by calculators. Instead, these problems are solved by a combination of programming and computational thinking. This course aims to develop the students' skills in programming and computational thinking to provide them with the tools they need to solve modern day engineering problems.

Consultation

Internal consultation

Internal Consultation

Consultants

None specified

Details

An initial draft of the proposal was submitted to Engineering Program Committee (EPC) and was tabled at the EPC meeting on 18 Nov 2016. The proposal proponent was asked to consult the Schools within the Faculty on this proposal and to report back to the EPC by April 2017.

All the Schools within the Faculty were consulted. The initial proposal (which proposed to focus exclusively on Python) received overwhelming support. Some Schools suggested that that minor components on Matlab and spreadsheet to be included in the course. This leads to the final form of proposal.

A more detailed report on the consultation can be found in the attachment.

Attachments

ENGG1811-consultation-report-AIMS.pdf

External consultation

External Consultation	**Consultants**	None specified
	Details	None specified
	Attachments	None specified
Interested Parties	Not specified	

Related Proposals

Related Proposals	Not specified

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)	Computing for Engineers
Student System ID	00057663
Can course be taken as General Education elective?	Yes
Field of Education	039999 – Engineering and Related Technologies not elsewhere classified

Course Review	
Next course review date	May 01, 2020
Provide details of any particular factors that need to be considered at that review.	Not specified

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	**Yes**	**Yes**
2018	No	**Yes**	**Yes**
2019	No	**Yes**	**Yes**
2020	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	0
Laboratory	2
Web-based Online Learning Activity	0.5
Clinical/Fieldwork	0
Distance Learning	0
Seminar	0
Studio	0
Meeting/Consultation	0
Total Hours per week	**5.5**

Primary delivery mode	Classroom
Secondary delivery modes	Online
Additional information about the delivery modes for this course	Part of the course may take the form of online modules. Students may also be asked to watch videos before attending the lectures.

Staff

Staff associated with course

Course Convenor

Name	Email	Role
Chun Tung Chou	ctchou@cse.unsw.edu.au	Associate Professor, School of Computer Science and Engineering

Administrative Contact	Name	Email	Role
	Julia Ciano	juliac@unsw.edu.au	–

Supplementary Information:

Resources

Student Resources	
Prescribed Resources	None specified
Recommended Resources	None specified

Experience and Assumed Knowledge

Industrial Experience Component	
Industrial Experience Component	Not specified

Assumed Knowledge	
Assumed Knowledge	The assumed knowledge on mathematics is the same as the current assumed knowledge of engineering undergraduate students. Later part of the course will require some familiarity with matrix operation but since most students would either take MATH1311 concurrently or have taken it, this should not be an issue.

Academic Structure:

Academic Structure

Prerequisites	
Prerequisite courses	Not specified
Prerequisite programs	Not specified
Prerequisite streams	Not specified
Prerequisite conditions	Not enrolled in any CSE major (streams: COMPxx, BINFxx, SENGxx)

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	Laboratory exercises	Lab Work	10%
	Assessment Description:	Satisfactory completion of prescribed exercises. Best 10 of 12 count.	
2	Programming assignments	Assignment	20%
	Assessment Description:	Two programming assignments using Python. Accessed objectively on performance plus conformation to design standards. Results for assignment, including report detailing deficiencies in the work, will be provided to the students.	
3	Mid-term test	Examination	10%
	Assessment Description:	Practical test in the laboratory. The test will cover the first 5 weeks of the course and students are expected to write Python programs in the test.	
4	Final exam	Examination	60%
	Assessment Description:	Practical exam in the laboratory. The exam will consists of multiple choice questions and programming questions.	
Total Weight			**100%**

Laboratory exercises

- Understand fundamental syntax and semantics of the Python programming language.
- Design, implement, test and debug complete Python programs to solve specified problems
- Understand and apply good practice for program organisation and programming style.
- Use spreadsheet applications to solve simple computational problems in engineering and science
- Use Matlab to solve simple computational problems in engineering and science

Programming assignments

- Understand fundamental syntax and semantics of the Python programming language.
- Design, implement, test and debug complete Python programs to solve specified problems
- Understand and apply good practice for program organisation and programming style.

Mid-term test

- Understand fundamental syntax and semantics of the Python programming language.
- Design, implement, test and debug complete Python programs to solve specified problems
- Understand and apply good practice for program organisation and programming style.

Final exam

- Understand fundamental syntax and semantics of the Python programming language.
- Design, implement, test and debug complete Python programs to solve specified problems
- Understand and apply good practice for program organisation and programming style.
- Use spreadsheet applications to solve simple computational problems in engineering and science
- Use Matlab to solve simple computational problems in engineering and science

Curriculum Mapping

Course Learning Outcomes

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

1	Understand fundamental syntax and semantics of the Python programming language.
2	Design, implement, test and debug complete Python programs to solve specified problems
3	Understand and apply good practice for program organisation and programming style.
4	Use spreadsheet applications to solve simple computational problems in engineering and science
5	Use Matlab to solve simple computational problems in engineering and science

Teaching strategies and Rationale

Teaching Strategies and Rationale

Lectures will be used to present the theory and practice of the techniques and tools in this course. There will be extensive use of practical demonstrations during lectures. Examples are real-world where possible, drawn from many disciplines of engineering but without requiring domain knowledge that students do not yet have.

The laboratory program fosters skills development in programming and problem solving. Lab classes are relatively small (18) and tutors encourage discussion within groups or across the cohort via online forums.

Course Aims

Course Aims	The course aims to equip students with the skills required to use programming and computational thinking to solve problems in engineering and related areas, and to assists students in developing familiarity with tools they will use within their own disciplines in later courses.

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)	The objective of this course is for students to acquire computing skills for solving problems in engineering. The course will develop the students' proficiency in a high level programming language and in using programming for problem solving. Topics: algorithms, program structure (statements, selection, iteration, functions), data types, arrays and matrices, reading and writing files, testing, code quality, simulation, animation, visualisation. The course includes 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).	Problem solving Programming