Handbook

COMP10001 Foundations of Computing

On this page:

Print - /view/2016/COMP10001?output=PDF

Credit Points: 12.5

Level: 1 (Undergraduate)

Dates & Locations:

This subject has the following teaching availabilities in 2016:

Semester 1, Parkville - Taught on campus. Show/hide details - #

Pre-teaching Period Start	not applicable
Teaching Period	29-Feb-2016 to 29-May-2016
Assessment Period End	24-Jun-2016
Last date to Self-Enrol	11-Mar-2016
Census Date	31-Mar-2016
Last date to Withdraw without fail	06-May-2016

Semester 2, Parkville - Taught on campus. Show/hide details - #

Pre-teaching Period Start	not applicable
Teaching Period	25-Jul-2016 to 23-Oct-2016
Assessment Period End	18-Nov-2016
Last date to Self-Enrol	05-Aug-2016
Census Date	31-Aug-2016
Last date to Withdraw without fail	23-Sep-2016

Time Commitment:

Timetable can be viewed here - https://sws.unimelb.edu.au/2016/Reports/List.aspx?objects=COMP10001&weeks=1-52&days=1-7&periods=1-56&template=module_by_group_list.
For information about these dates, click here - http://ask.unimelb.edu.au/app/answers/detail/a_id/6032.
Contact Hours: 60 hours, comprised of three 1-hour lectures and one 2-hour workshop per week
Total Time Commitment:

170 hours

Prerequisites: None

Corequisites: None

Recommended Background Knowledge: None

Subject
INFO10001 Informatics 1: Data on the Web - /view/2016/INFO10001

Non Allowed Subjects:	INFO10001 Informatics-1:Practical Computing (prior to 2011) 615-145 Concepts of Software Development 1 433-151 Introduction to Programming (Advanced) 433-171 Introduction to Programming 600-151 Informatics-1: Practical Computing
Core Participation Requirements:	For the purposes of considering request for Reasonable Adjustments under the Disability Standards for Education (Cwth 2005), and Student Support and Engagement Policy, academic requirements for this subject are articulated in the Subject Overview, Learning Outcomes, Assessment and Generic Skills sections of this entry. It is University policy to take all reasonable steps to minimise the impact of disability upon academic study, and reasonable adjustments will be made to enhance a student's participation in the University's programs. Students who feel their disability may impact on meeting the requirements of this subject are encouraged to discuss this matter with a Faculty Student Adviser and Student Equity and Disability Support: http://services.unimelb.edu.au/disability - http://services.unimelb.edu.au/disability

Coordinator

Prof Christopher Leckie, Prof Tim Baldwin

Contact

Semester 1: Professor Tim Baldwin

email: tbaldwin@unimelb.edu.au - mailto:tbaldwin@unimelb.edu.au

Semester 2: A/Prof Chris Leckie

email: caleckie@unimelb.edu.au - mailto:bjpope@unimelb.edu.au

Subject Overview:	**AIMS** Solving problems in areas such as business, biology, physics, chemistry, engineering, humanities, and social sciences often requires manipulating, analysing, and visualising data through computer programming. This subject teaches students with little or no background in computer programming how to design and write basic programs using a high-level procedural programming language, and to solve simple problems using these skills. This subject is the first subject in the Computing & Software Systems and the Informatics majors, and introduces students to programming and the basics of algorithmic thinking. **INDICATIVE CONTENT** Fundamental programming constructs; fundamental data structures; abstraction; basic program structures; algorithmic problem solving, testing and debugging; introduction to the Web, multimedia and visualisation. Examples of projects that students complete are: • A text analytics “library” consisting of a series of independent functions to calculate/extract different things given a document/document collection as input • A video recommender system, broken down into a series of functions • An AI player for an online card game, designed such that students play off against each other (and against the class) at the end of semester.

Learning Outcomes:	**INTENDED LEARNING OUTCOMES (ILO)** On completion of this subject the student is expected to: 1. Use the fundamental programming constructs (sequence, alternation, selection) 2. Use the fundamental data structures (arrays, records, lists, associative arrays) 3. Use abstraction constructs such as functions 4. Understand and employ some basic program structures 5. Understand and employ some basic algorithmic problem solving techniques 6. Read, write, and debug simple, small programs
Assessment:	• A three-stage project, requiring approximately 30 - 35 hours of work, with stages due at the end of each third of the semester - approximately weeks 4, 8, and 12 (30%) • One 1-hour mid-semester test (10%) • A workshop assignment to demonstrate programming competency, due two thirds of the way through semester (10%), requiring approximately 10 - 13 hours of work per student • One 2-hour end-of-semester examination (50%). **Hurdle requirement:** To pass the subject, students must obtain at least: • 50% overall, 20/40 for the project and assignment work • And 30/60 for the mid-semester test and end-of-semester written examination combined. Intended Learning Outcomes (ILOs) 1-6 are addressed in the projects, the mid-semester test, and the workshop assignment and the final exam.
Prescribed Texts:	None
Breadth Options:	This subject potentially can be taken as a breadth subject component for the following courses: • Bachelor of Arts - https://handbook.unimelb.edu.au/view/2016/B-ARTS • Bachelor of Commerce - https://handbook.unimelb.edu.au/view/2016/B-COM • Bachelor of Environments - https://handbook.unimelb.edu.au/view/2016/B-ENVS • Bachelor of Music - https://handbook.unimelb.edu.au/view/2016/B-MUS You should visit learn more about breadth subjects - http://breadth.unimelb.edu.au/breadth/info/index.html and read the breadth requirements for your degree, and should discuss your choice with your student adviser, before deciding on your subjects.
Fees Information:	Subject EFTSL, Level, Discipline & Census Date - http://enrolment.unimelb.edu.au/fees
Generic Skills:	On completion of this subject, students should have developed the following generic skills: • An ability to apply knowledge of basic science and engineering fundamentals • An ability to undertake problem identification, formulation and solution • The capacity to solve problems, including the collection and evaluation of information • The capacity for critical and independent thought and reflection • An expectation of the need to undertake lifelong learning, and the capacity to do so.
	LEARNING AND TEACHING METHODS The subject is delivered through a combination of lectures and workshops (combination of tutorial and individual/group work in a computer lab). Students get

Notes:	a hands-on introduction to Python through a series of online worksheets with embedded programming tasks/automatic assessment, and then go on to complete three projects. **INDICATIVE KEY LEARNING RESOURCES** Students have access to lecture notes, lecture slides, tutorial worksheets, which houses the interactive worksheets as well as a programming environment. The subject LMS site also contains links to recommended resources relating to basic programming, and advanced problems for students who want to extend themselves. **CAREERS / INDUSTRY LINKS** As an introductory programming subject, this is relevant to all aspects of the IT industry. Exemplar companies/organisations which have been involved in the delivery of the subject (through guest lectures etc.) are: Palantir Technologies (software engineering, intelligent systems), AURIN (Australian Urban Research Infrastructure Network: geomatics, distributed computing, web development), VLSCI (Victorian Life Sciences Computing Initiative; computational biology, bioinformatics, distributed computing, big data). There have also been guest lecturers from within the university in fields including computational ophthalmology, electronic voting, and social media analysis.
Related Course(s):	Bachelor of Biomedicine - /view/2016/B-BMED Diploma in Informatics - /view/2016/D-INFO
Related Majors/Minors/Specialisations:	Science-credited subjects - new generation B-SCI and B-ENG. - /view/2016/%21B-SCI-SPC%2B1021 Selective subjects for B-BMED - /view/2016/%21B-BMED-SPC%2B1000

Date created: *11 October 2007* **Last modified:** *22 January 2009* **Authoriser:** *Vice-Principal and Academic Registrar, Office of the Vice-Principal and Academic Registrar* **Maintainer:** *Handbook Team, Applications Management, Academic Services Applications Development* **Help:** *for Future Students* *for Current Students*