

COMP10002 Foundations of Algorithms

Credit Points:	12.50								
Level:	1 (Undergraduate)								
Dates & Locations:	This subject is not offered in 2013.								
Time Commitment:	Contact Hours: 60 hours, comprising of three 1-hour lectures and one 2-hour workshop per week Total Time Commitment: 120 hours								
Prerequisites:	<table border="1"> <thead> <tr> <th>Subject</th> <th>Study Period Commencement:</th> <th>Credit Points:</th> </tr> </thead> <tbody> <tr> <td>COMP10001 Foundations of Computing</td> <td>Semester 2</td> <td>12.50</td> </tr> </tbody> </table> <p>OR</p> <p>Achieving at least 75% in a programming competency test.</p>			Subject	Study Period Commencement:	Credit Points:	COMP10001 Foundations of Computing	Semester 2	12.50
Subject	Study Period Commencement:	Credit Points:							
COMP10001 Foundations of Computing	Semester 2	12.50							
Corequisites:	None								
Recommended Background Knowledge:	None								
Non Allowed Subjects:	<p>Students cannot enrol in and gain credit for this subject and:</p> <table border="1"> <thead> <tr> <th>Subject</th> <th>Study Period Commencement:</th> <th>Credit Points:</th> </tr> </thead> <tbody> <tr> <td>COMP20006 Programming the Machine</td> <td>Not offered 2013</td> <td>12.50</td> </tr> </tbody> </table> <p>433-172 Algorithmic Problem Solving 433-152 Algorithmic Problem Solving</p>			Subject	Study Period Commencement:	Credit Points:	COMP20006 Programming the Machine	Not offered 2013	12.50
Subject	Study Period Commencement:	Credit Points:							
COMP20006 Programming the Machine	Not offered 2013	12.50							
Core Participation Requirements:	<p><p>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.</p> <p>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</p></p>								
Contact:	<p>Alistair Moffat email: ammoffat@unimelb.edu.au (mailto:ammoffat@unimelb.edu.au) Peter Stucky email: p.stuckey@unimelb.edu.au (mailto:p.stuckey@unimelb.edu.au)</p>								
Subject Overview:	<p>In many projects, it is important for programmers to have fine control over low-level details of program execution, and to be able to assess the cost of a design decision on likely overall program performance. This subject introduces students to a system programming language that gives programmers this kind of control, explores a range of standard data structures and algorithmic techniques, and shows how to apply them to frequently encountered problems.</p>								
Objectives:	<p>On completion of this subject, students should be able to:</p> <ul style="list-style-type: none"> # Read, write and debug typical small-scale programs in a system programming language such as C. # Discuss, analyze, implement, and apply standard data structures such as linked lists, binary search trees, and hash tables. 								

	<ul style="list-style-type: none"> # Discuss, analyze, and apply a range of sorting and searching algorithms. # Present logical arguments for the correctness of a given algorithm. # Choose between different algorithms for simple problems by analysing their complexity. # Use a command line interface and standard development tools for programming.
Assessment:	Project work during semester, expected to take about 36 hours (30%), due in approximately Week 8 and Week 11 One mid-semester test (10%), held in Week 5 or Week 6 One two-hour end-of-semester examination, including a practical programming component (60%) To pass the subject, students must obtain at least: 50% overall; 12/30 in project work; 28/70 in the mid-semester test and end-of-semester written examination combined. ILO1 is addressed in all components of assessment. ILO2, ILO3, ILO4 and ILO5 are assessed in the mid-semester test and in the examination. ILO6 is addressed in the programming assignments.
Prescribed Texts:	None
Breadth Options:	<p>This subject potentially can be taken as a breadth subject component for the following courses:</p> <ul style="list-style-type: none"> # Bachelor of Arts (https://handbook.unimelb.edu.au/view/2013/B-ARTS) # Bachelor of Commerce (https://handbook.unimelb.edu.au/view/2013/B-COM) # Bachelor of Environments (https://handbook.unimelb.edu.au/view/2013/B-ENVS) # Bachelor of Music (https://handbook.unimelb.edu.au/view/2013/B-MUS) <p>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.</p>
Fees Information:	Subject EFTSL, Level, Discipline & Census Date, http://enrolment.unimelb.edu.au/fees
Generic Skills:	<p>On completion of this subject students should have:</p> <ul style="list-style-type: none"> # Ability to undertake problem identification, formulation and solution # Capacity for independent critical thought, rational inquiry and self-directed learning # Profound respect for truth and intellectual integrity, and for the ethics of scholarship # An ability to apply knowledge of basic science and engineering fundamentals
Notes:	This subject is available for science credit to students enrolled in the BSc (both pre-2008 and new degrees), BAsc or a combined BSc course. Students undertaking this subject will be expected to regularly access an internet-enabled computer.
Related Majors/Minors/Specialisations:	Science credit subjects* for pre-2008 BSc, BAsc and combined degree science courses Science-credited subjects - new generation B-SCI and B-ENG. Core selective subjects for B-BMED.
Related Breadth Track(s):	Computing