

COMP10002 Foundations of Algorithms

Credit Points:	12.5						
Level:	1 (Undergraduate)						
Dates & Locations:	2016, Parkville This subject commences in the following study period/s: Semester 1, Parkville - Taught on campus. Semester 2, Parkville - Taught on campus.						
Time Commitment:	Contact Hours: 60 hours, comprising of three 1-hour lectures and one 2-hour workshop per week Total Time Commitment: 170 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 1, 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 1, Semester 2	12.50
Subject	Study Period Commencement:	Credit Points:					
COMP10001 Foundations of Computing	Semester 1, 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 2016</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 2016	12.50
Subject	Study Period Commencement:	Credit Points:					
COMP20006 Programming the Machine	Not offered 2016	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>						
Coordinator:	Dr Jianzhong Qi, Prof Alistair Moffat						
Contact:	<p>Semester 1: Dr Jianzhong Qi email: jianzhong.qi@unimelb.edu.au (mailto:jianzhong.qi@unimelb.edu.au)</p> <p>Semester 2: Professor Alistair Moffat email: ammoffat@unimelb.edu.au (mailto:ammoffat@unimelb.edu.au)</p>						
Subject Overview:	AIMS						

	<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> <p>INDICATIVE CONTENT</p> <ul style="list-style-type: none"> # Introduction to computer organization # Machine level representation of data # Programming in a system programming language # Pointers and dynamic memory allocation # Program semantics and arguments about correctness # Induction and recursion # Basic searching algorithms (linear and binary) # Basic sorting algorithms (such as selection sort, insertion sort, quicksort) # Basic data structures (binary search trees and hash tables) # Asymptotic complexity # Standard software development tools such as debuggers.
<p>Learning Outcomes:</p>	<p>INTENDED LEARNING OUTCOMES (ILO)</p> <p>On completion of this subject the student is expected to:</p> <ol style="list-style-type: none"> 1 Read, write and debug typical small-scale programs in a system programming language such as C 2 Discuss, analyse, implement, and apply standard data structures such as linked lists, binary search trees, and hash tables 3 Discuss, analyse, and apply a range of sorting and searching algorithms 4 Present logical arguments for the correctness of a given algorithm 5 Choose between different algorithms for simple problems by analysing their complexity 6 Use a command line interface and standard development tools for programming
<p>Assessment:</p>	<p>Project work during semester, requiring approximately 30 - 35 hours of work, (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 (60%). Hurdle requirement: 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. Intended Learning Outcomes (ILO) 1 is addressed in all components of assessment. ILO 2, ILO 3, ILO 4 and ILO 5 are assessed in the mid-semester test and in the examination. ILO 6 is addressed in the programming assignments.</p>
<p>Prescribed Texts:</p>	<p>None</p>
<p>Breadth Options:</p>	<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/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) <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>
<p>Fees Information:</p>	<p>Subject EFTSL, Level, Discipline & Census Date, http://enrolment.unimelb.edu.au/fees</p>
<p>Generic Skills:</p>	<p>On completion of this subject students should have developed the following skills:</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:	<p>This subject is available for science credit to students enrolled in the BSc. Students undertaking this subject will require regular access to the internet.</p> <p>LEARNING AND TEACHING METHODS</p> <p>The subject will be delivered through a combination of lectures, programming workshops, and programming exercises and assessed programming assignments. Students will also be expected to develop and submit programming assignments for assessment.</p> <p>INDICATIVE KEY LEARNING RESOURCES</p> <p>Students will have access to lecture notes and lecture slides, and will be expected to own a copy of the textbook nominated by the coordinator. Other guidance will be provided via the LMS.</p> <p>CAREERS / INDUSTRY LINKS</p> <p>Programming competencies are a critical part of a range of career pathways, including in IT, engineering, and mathematics. In addition, an understanding of the fundamental nature of computation is a key enabler across all science discipline areas, including genetics and physics.</p>
Related Majors/Minors/ Specialisations:	Science-credited subjects - new generation B-SCI and B-ENG. Selective subjects for B-BMED