COMP10002 Foundations of Algorithms

<table>
<tr><td><b>Credit Points:</b></td><td>12.50</td></tr>
<tr><td><b>Level:</b></td><td>1 (Undergraduate)</td></tr>
<tr><td><b>Dates &amp; Locations:</b></td><td>2012, Parkville
This subject commences in the following study period/s:
Semester 2, Parkville - Taught on campus.
On campus only</td></tr>
<tr><td><b>Time Commitment:</b></td><td>Contact Hours: 36 one-hour lectures (three per week) and 12 two-hour workshops (one per week) Total Time Commitment: 120 hours</td></tr>
<tr><td><b>Prerequisites:</b></td><td><table><tr><th>Subject</th><th>Study Period Commencement:</th><th>Credit Points:</th></tr><tr><td>COMP10001 Foundations of Computing</td><td>Semester 1, Semester 2</td><td>12.50</td></tr></table>OR
Achieving at least 75% in a programming competency test.</td></tr>
<tr><td><b>Corequisites:</b></td><td>None</td></tr>
<tr><td><b>Recommended Background Knowledge:</b></td><td>None</td></tr>
<tr><td><b>Non Allowed Subjects:</b></td><td><table><tr><th>Subject</th><th>Study Period Commencement:</th><th>Credit Points:</th></tr><tr><td>COMP20006 Programming the Machine</td><td>Semester 1, Semester 2</td><td>12.50</td></tr></table>433-172 Algorithmic Problem Solving
433 152-Algorithmic Problem Solving</td></tr>
<tr><td><b>Core Participation Requirements:</b></td><td>&lt;p&gt;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: <a href="http://services.unimelb.edu.au/disability">http://services.unimelb.edu.au/disability</a>&lt;/p&gt;</td></tr>
<tr><td><b>Coordinator:</b></td><td>Prof Alistair Moffat</td></tr>
<tr><td><b>Contact:</b></td><td>Associate Professor Tim Baldwin
email:<b>tbaldwin@unimelb.edu.au (mailto:TBALDWIN@UNIMELB.EDU.AU)</b></td></tr>
<tr><td><b>Subject Overview:</b></td><td>In many projects, it is important for programmers to have fine control over low-level details of program execution. This subject introduces students to a system programming language that gives programmers this kind of control, and shows how to apply standard algorithmic solutions to frequently encountered problems. Topics covered include: introduction to computer organization; machine level representation of data; programming in a system programming language; pointers and dynamic memory allocation; propositional logic; induction and recursion; basic searching algorithms (linear and binary); basic sorting algorithms (such as selection</td></tr>
</table>

	sort, insertion sort or quicksort); basic data structures (binary search trees and hash tables); asymptotic complexity; standard software development tools such as debuggers.
Objectives:	On completion of this subject, students should be able to: # Read, write and debug simple sorting and searching algorithms # Read, write and debug code using standard data structures such as binary search trees and hash tables # Present arguments for the correctness or incorrectness of a given algorithm # Choose between different algorithms for simple problems by analysing their complexity # Read, write and debug typical small scale programs in a system programming language such as C # Use a command line interface for programming.
Assessment:	Project work during semester, expected to take about 36 hours (30%) A mid-semester test (10%) A 3-hour end-of-semester written examination (60%) To pass the subject, students must obtain at least 50% overall 15/30 in project work And 35/70 in the mid-semester test and end-of-semester written examination combined.
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/2012/B-ARTS)** # **Bachelor of Commerce (https://handbook.unimelb.edu.au/view/2012/B-COM)** # **Bachelor of Music (https://handbook.unimelb.edu.au/view/2012/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 the: # 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 through 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
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