

COMP20007 Design of Algorithms

Credit Points:	12.50									
Level:	2 (Undergraduate)									
Dates & Locations:	This subject is not offered in 2014.									
Time Commitment:	Contact Hours: 48 hours, comprising of two 1-hour lectures and one 2-hour workshop per week Total Time Commitment: 170 hours									
Prerequisites:	<p>25 points of university-level Mathematics</p> <p>AND</p> <table border="1"> <thead> <tr> <th>Subject</th> <th>Study Period Commencement:</th> <th>Credit Points:</th> </tr> </thead> <tbody> <tr> <td>COMP10002 Foundations of Algorithms</td> <td>Semester 1, Semester 2</td> <td>12.50</td> </tr> </tbody> </table>	Subject	Study Period Commencement:	Credit Points:	COMP10002 Foundations of Algorithms	Semester 1, Semester 2	12.50			
Subject	Study Period Commencement:	Credit Points:								
COMP10002 Foundations of Algorithms	Semester 1, Semester 2	12.50								
Corequisites:	None									
Recommended Background Knowledge:	None									
Non Allowed Subjects:	<table border="1"> <thead> <tr> <th>Subject</th> <th>Study Period Commencement:</th> <th>Credit Points:</th> </tr> </thead> <tbody> <tr> <td>COMP20003 Algorithms and Data Structures</td> <td>Semester 2</td> <td>12.50</td> </tr> <tr> <td>COMP90038 Algorithms and Complexity</td> <td>Semester 1, Semester 2</td> <td>12.50</td> </tr> </tbody> </table> <p>433-253 Algorithms and Data Structures 433 298 Algorithms and Data Structures</p>	Subject	Study Period Commencement:	Credit Points:	COMP20003 Algorithms and Data Structures	Semester 2	12.50	COMP90038 Algorithms and Complexity	Semester 1, Semester 2	12.50
Subject	Study Period Commencement:	Credit Points:								
COMP20003 Algorithms and Data Structures	Semester 2	12.50								
COMP90038 Algorithms and Complexity	Semester 1, Semester 2	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:	email: aturpin@unimelb.edu.au (mailto:aturpin@unimelb.edu.au)									
Subject Overview:	<p>AIMS</p> <p>Programmers can choose between several representations of data. These will have different strengths and weaknesses, and each will require its own set of algorithms. This subject will cover some of the most frequently used data structures and their associated algorithms. The emphasis will be on justification of algorithm correctness, on analysis of algorithm performance, and on choosing the right data structure for the problem at hand.</p> <p>INDICATIVE CONTENT</p> <p>Sample projects are: approximate string matching for a translation memory, involving sorting and comparison of dynamic programming, branch-and-bound search and brute-force search using a variety of data structures (e.g. arrays, hash tables, tries); speech synthesis based on a pronouncing dictionary and pre-prepared grapheme-phoneme alignment data, based on parsing of the alignment data, hashing of variable length n-grams, and a variety of models for predicting</p>									

	the phoneme(s) associated with a given phoneme sequence (e.g. a simple unigram baseline or a hidden Markov model).
Learning Outcomes:	<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 graph algorithms, advanced sorting algorithms, dynamic programs and greedy algorithms 2 Read, write and debug code using intermediate data structures 3 Choose between different algorithms for intermediate problems by analysing their complexity 4 Read, write and debug typical multi-module programs in a system programming language such as C
Assessment:	A two stage project work during semester, expected to take about 36 hours, with the first stage due in week 4 and the second stage due in week 10 (30%) One 45 minute mid-semester test (10%) One 3-hour end-of-semester written examination (60%) Hurdle requirement: 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. Intended Learning Outcomes (ILOs) 1 - 4 are addressed through the projects and in workshops. ILOs 1 - 3 are assessed in the mid-semester test and the final exam.
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/2014/B-ARTS) # Bachelor of Commerce (https://handbook.unimelb.edu.au/view/2014/B-COM) # Bachelor of Environments (https://handbook.unimelb.edu.au/view/2014/B-ENVS) # Bachelor of Music (https://handbook.unimelb.edu.au/view/2014/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 developed the following skills:</p> <ul style="list-style-type: none"> # 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
Notes:	<p>LEARNING AND TEACHING METHODS</p> <p>The subject is delivered through a combination of lectures and workshops (combination of tutorial and individual/group work in a computer lab). The subject consolidates the introductory C programming exposure students will have attained through COMP10002 Foundations of Algorithms or COMP20006 Programming the Machine.</p> <p>INDICATIVE KEY LEARNING RESOURCES</p> <p>Students have access to lecture notes, lecture slides and tutorial worksheets. The subject LMS site also contains links to recommended resources relating to programming and algorithmics, and advanced problems for students who want to extend themselves.</p> <p>CAREERS / INDUSTRY LINKS</p> <p>The subject provides students with intermediate-level, multi-module programming experience in C, and hands-on practicum with algorithmic analysis and development. It is highly relevant to any career involving “big data”, and covers the staple of technical interviews by companies such</p>

	as Google, Microsoft and Amazon. There is every expectation that the subject will incorporate guest lecture(s) from leading figures in algorithmics.
Related Majors/Minors/ Specialisations:	Computer Science Science credit subjects* for pre-2008 BSc, BAsC and combined degree science courses Science-credited subjects - new generation B-SCI and B-ENG. Selective subjects for B-BMED
Related Breadth Track(s):	Computing