

COMP30026 Models of Computation

Credit Points:	12.5						
Level:	3 (Undergraduate)						
Dates & Locations:	2015, Parkville This subject commences in the following study period/s: Semester 2, Parkville - Taught on campus.						
Time Commitment:	Contact Hours: 48 hours, Two, one-hour lectures per week and one, two-hour tutorial per week. Total Time Commitment: 170 hours						
Prerequisites:	One first year mathematics subject (12.5 points) OR Admission to the MC-ENG Master of Engineering (Software) or (Software with Business) or MC-IT Master of Information Technology						
Corequisites:	None						
Recommended Background Knowledge:	None						
Non Allowed Subjects:	Students cannot enrol in and gain credit for this subject and: 433-255 Logic and Computation <table border="1" data-bbox="387 992 1485 1137"> <thead> <tr> <th>Subject</th> <th>Study Period Commencement:</th> <th>Credit Points:</th> </tr> </thead> <tbody> <tr> <td>COMP20004 Discrete Structures</td> <td>Not offered 2015</td> <td>12.50</td> </tr> </tbody> </table>	Subject	Study Period Commencement:	Credit Points:	COMP20004 Discrete Structures	Not offered 2015	12.50
Subject	Study Period Commencement:	Credit Points:					
COMP20004 Discrete Structures	Not offered 2015	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><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> </p>						
Coordinator:	Assoc Prof Harald Sondergaard						
Contact:	email: harald@unimelb.edu.au (mailto:harald@unimelb.edu.au)						
Subject Overview:	<p>AIMS</p> <p>Formal logic and discrete mathematics provide the theoretical foundations for computer science. This subject uses logic and discrete mathematics to model the science of computing. It provides a grounding in the theories of logic, sets, relations, functions, automata, formal languages, and computability, providing concepts that underpin virtually all the practical tools contributed by the discipline, for automated storage, retrieval, manipulation and communication of data.</p> <p>INDICATIVE CONTENT</p> <ul style="list-style-type: none"> # Logic: Propositional and predicate logic, resolution proofs, mathematical proof # Discrete mathematics: Sets, functions, relations, order, well-foundedness, induction and recursion # Automata: Regular languages, finite-state automata, context-free grammars and languages, parsing 						

	# Computability: Turing machines, reducibility, decidability.
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 Use propositional and predicate logic as tools to reason about non-trivial computational problems 2 Explain basic principles of mechanised reasoning, including resolution proof, and apply these to reason about computational problems 3 Reason about properties of mathematical objects such as functions and relations, and apply them to computational problems 4 Apply discrete mathematical techniques to problems in computer science 5 Synthesize context-free grammars from less formal language specifications 6 Design abstract computational devices, from finite-state automata to Turing machines 7 Analyze and reason about computational models, including finite-state automata, regular expressions, context-free grammars, and Turing machines 8 Explain the concepts of computability and decidability 9 Use problem reducibility to establish the decidability or otherwise of simple problems.
Assessment:	One individual 800-word project, requiring approximately 13 - 15 hours of work, due in week 6 (10%). Addresses Intended Learning Outcomes (ILOs) 1 and 2 One 45-minute mid-semester written test, due in week 7 (10%). Addresses ILOs 1, 2 and 3 One individual 800-word project, requiring approximately 13 - 15 hours of work, due in week 11 (10%). Addresses ILOs 3, 4, 5, and 6 One 2-hour end-of-semester written examination (70%). Addresses all ILOs.
Prescribed Texts:	None
Breadth Options:	This subject is not available as a breadth subject.
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"> # Analytical skills # Reasoning and problem-solving skills # Ability to communicate with precision, rigour and efficacy # Ability to apply knowledge of science and engineering fundamentals # Capacity for creativity and innovation # Ability to undertake problem identification, formulation and solution.
Notes:	<p>LEARNING AND TEACHING METHODS</p> <p>The subject involves two 1-hour lectures per week followed by a 2-hour tutorial. Weekly tutorial problems are assigned and discussed in class. Tutors use some tutorial time to demonstrate applications of the theory, such as SAT-solver use, logic programming, and parsing. Lectures and tutorials are designed to be highly interactive, and the written assignments are designed to be challenging, so as to generate discussion. Although written assignments are submitted by students individually, in-plenum discussion of the problems is allowed, and encouraged.</p> <p>INDICATIVE KEY LEARNING RESOURCES</p> <p>The subject uses online reading materials and offers access to visualisation tools (the JFLAP suite), an online discussion forum, and advance access to all teaching materials, including slides used in lectures.</p> <p>CAREERS / INDUSTRY LINKS</p> <p>The subject is foundational. While the practice of computing changes fast, the theoretical underpinnings, and many of the basic concepts underlying computation, change only slowly. A foundation in logic and mathematics provides important conceptual tools that are used by theoreticians, computer scientists, and software engineering practitioners alike.</p>
Related Course(s):	Master of Information Technology
Related Majors/Minors/Specialisations:	B-ENG Software Engineering stream Computer Science

Computer Science
Computer Science
Computer Science
Computing and Software Systems
MIT Computing Specialisation
Master of Engineering (Software with Business)
Master of Engineering (Software)
Science-credited subjects - new generation B-SCI and B-ENG.