COMP90057 Advanced Theoretical Computer Science

<table>
<tr><td><b>Credit Points:</b></td><td>12.5</td></tr>
<tr><td><b>Level:</b></td><td>9 (Graduate/Postgraduate)</td></tr>
<tr><td><b>Dates &amp; Locations:</b></td><td>2016, Parkville
This subject commences in the following study period/s:
Semester 2, Parkville - Taught on campus.</td></tr>
<tr><td><b>Time Commitment:</b></td><td>Contact Hours: 48 hours Total Time Commitment: 200 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>COMP30026 Models of Computation</td><td>Semester 2</td><td>12.50</td></tr>
</table>
OR Equivalent
(COMP20004 Discrete Structures prior to 2014)</td></tr>
<tr><td><b>Corequisites:</b></td><td>None</td></tr>
<tr><td><b>Recommended Background Knowledge:</b></td><td>Proficiency in discrete mathematics and propositional logic.</td></tr>
<tr><td><b>Non Allowed Subjects:</b></td><td>433-330 Theory of Computation
<table>
<tr><th>Subject</th><th>Study Period Commencement:</th><th>Credit Points:</th></tr>
<tr><td>COMP30025 Theory of Computation</td><td>Not offered 2016</td><td>12.50</td></tr>
<tr><td>COMP30021 Theoretical Computer Science</td><td>Not offered 2016</td><td>12.50</td></tr>
</table></td></tr>
<tr><td><b>Core Participation Requirements:</b></td><td><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: <a href="http://services.unimelb.edu.au/disability">http://services.unimelb.edu.au/disability</a></p></td></tr>
<tr><td><b>Coordinator:</b></td><td>Assoc Prof Tony Wirth</td></tr>
<tr><td><b>Contact:</b></td><td><b>awirth@unimelb.edu.au (mailto:awirth@unimelb.edu.au)</b></td></tr>
<tr><td><b>Subject Overview:</b></td><td><b>AIMS</b>
At the heart of theoretical computer science are questions of both philosophical and practical importance. What does it mean for a problem to be solvable by computer? What are the limits of computability? Which types of problems can be solved efficiently? What are our options in the face of intractability? This subject covers such questions in the content of a wide-ranging exploration of the nexus between logic, complexity and algorithms, and examines many important (and sometimes surprising) results about the nature of computing.
<b>INDICATIVE CONTENT</b>
Turing machines
The Church-Turing Thesis</td></tr>
</table>

	# Decidable languages # Reducability # Time Complexity: The classes P and NP, NP-complete problems # Space complexity: including sub-linear space # Circuit complexity # Approximation algorithms # Probabilistic complexity classes # Additional topics may include descriptive complexity, interactive proofs, communication complexity, complexity as applied to cryptography # Space complexity, including sub-linear space # Finite state automata, pushdown automata, regular languages, context-free languages to the Recommended Background Knowledge. Example of assignment # Proving the equivalence of a variant of a standard machine to the original version # Describing an NP-hardness reduction # Designing an approximation algorithm for an NP-hard problem.
Learning Outcomes:	**INTENDED LEARNING OUTCOMES (ILO)** On completion of this subject the student is expected to: 1 Design, manipulate, and reason about Turing machines 2 Account for the inherent complexity of many computational problems of practical importance 3 Conduct formal reasoning about machines, circuits, problems and algorithms, including reduction-based proof 4 Design approximation algorithms for intractable problems 5 Apply complexity arguments to related fundamental computational problems, such as randomized computations, interactive proof systems and cryptographic pseudorandom generators
Assessment:	Three individual assignments involving mathematical proof and possibly some programming, requiring approximately 35 - 40 hours of work in total, due in weeks 4, 7 and 11 (30%) 3-hour written examination held at the end of semester (70%). Hurdle Requirement: To pass the subject, students must obtain at least: 15/30 in the assignments 35/70 on the examination. Assessment addresses all Intended Learning Outcomes (ILOs)
Prescribed Texts:	Michael Sipser, "Introduction to the Theory of Computation", 3rd Edition.
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:	On completion of this subject, students should have developed the following skills: # Ability to apply knowledge of science and engineering fundamentals # Ability to communicate effectively, with the engineering team and with the community at large # Capacity for lifelong learning and professional development # Profound respect for truth and intellectual integrity, and for the ethics of scholarship.
Related Course(s):	Doctor of Philosophy - Engineering Master of Information Technology Master of Information Technology Master of Philosophy - Engineering Master of Science (Computer Science)
Related Majors/Minors/ Specialisations:	MIT Computing Specialisation MIT Distributed Computing Specialisation MIT Health Specialisation MIT Spatial Specialisation

	Master of Engineering (Software with Business) Master of Engineering (Software)