COMP30025 Theory of Computation

Credit Points:	12.50
Level:	3 (Undergraduate)
Dates & Locations:	This subject is not offered in 2012.
Time Commitment:	Contact Hours: 24 one-hour lectures (two per week) and 12 two-hour workshops (one per week) Total Time Commitment: 120 hours
Prerequisites:	25 points of university-level mathematics
Corequisites:	None
Recommended Background Knowledge:	None
Non Allowed Subjects:	**Subject** / **Study Period Commencement:** / **Credit Points:** COMP30021 Theoretical Computer Science / Semester 2 / 12.50 433-330 Theory of Computation
Core Participation Requirements:	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. 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
Contact:	Associate Professor Tim Baldwin email: **tbaldwin@unimelb.edu.au (mailto:tbaldwin@unimelb.edu.au)**
Subject Overview:	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 context 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. Topics covered include: predicate logic; regular languages; finite-state automata; context-free grammars and languages; pumping lemmas; Turing machines; computability; the Church-Turing thesis; decidability; reducibility; complexity: the classes P and NP; NP-complete problems; space complexity.
Objectives:	On completion of this subject students should be able to: # Describe the limitations of computing devices # Analyse the inherent complexity of many computational problems of practical importance # Conduct formal reasoning about machines, problems and algorithms, including reduction-based proofs
Assessment:	Written assignments during semester, expected to take about 36 hours (30%) A 3-hour end-of-semester written examination (70%) To pass the subject, students must obtain at least 50% overall 15/30 in project work And 35/70 in the written examination

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 developed the following generic skills: # An ability to apply knowledge of basic science and engineering fundamentals # Capacity for creativity and innovation # An ability to undertake problem identification, formulation and solution # Capacity for lifelong learning and professional development
Related Majors/Minors/ Specialisations:	Computing and Software Systems