# COMP90038 Algorithms and Complexity

| | |
|---|---|
| **Credit Points:** | 12.50 |
| **Level:** | 9 (Graduate/Postgraduate) |
| **Dates & Locations:** | This subject is not offered in 2014. |
| **Time Commitment:** | Contact Hours: 36 hours, comprising of three hours of lectures per week Total Time Commitment: 200 hours |
| **Prerequisites:** | An undergraduate degree in a cognate discipline. |
| **Corequisites:** | None |
| **Recommended Background Knowledge:** | Basic proficiency in mathematics and computing. |
| **Non Allowed Subjects:** | |

| Subject | Study Period Commencement: | Credit Points: |
|---|---|---|
| COMP20003 Algorithms and Data Structures | Semester 2 | 12.50 |
| COMP20007 Design of Algorithms | Semester 1 | 12.50 |

| | |
|---|---|
| **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: <a href="http://services.unimelb.edu.au/disability">http://services.unimelb.edu.au/disability</a> |
| **Contact:** | email: **harald@unimelb.edu.au (mailto:harald@unimelb.edu.au)** |
| **Subject Overview:** | **AIMS**<br>The aim of this subject is for students to develop familiarity and competence in assessing and designing computer programs for computational efficiency. Although computers manipulate data very quickly, to solve large-scale problems, we must design strategies so that the calculations combine effectively. Over the latter half of the 20 th century, an elegant theory of computational efficiency developed. This subject introduces students to the fundamentals of this theory and to many of the classical algorithms and data structures that solve key computational questions. These questions include distance computations in networks, searching items in large collections, and sorting them in order.<br>**INDICATIVE CONTENT**<br>Topics covered include complexity classes and asymptotic notation; empirical analysis of algorithms; abstract data types including queues, trees, priority queues and graphs; algorithmic techniques including brute force, divide-and-conquer, dynamic programming and greedy approaches; space and time trade-offs; and the theoretical limits of algorithm power. |
| **Learning Outcomes:** | **INTENDED LEARNING OUTCOMES (ILO)**<br>On completion of this subject the student is expected to:<br>1 Design, manipulate and reason about a variety of techniques for solving sorting, searching and graph problems<br>2 Write efficient algorithms and data structures for a variety of fundamental problems<br>3 Conduct formal reasoning about problem complexity and algorithmic efficiency |

| | |
|---|---|
| | 4 Recognize the design techniques of standard algorithms, and apply these techniques to develop new computational solutions to problems |
| **Assessment:** | Project work during semester, due around weeks 6 and 11, expected to take approximately up to 48 hours (20%) A 45 minute mid-semester test, around week 7 (10%) A 3-hour end-of-semester written examination (70%) Hurdle requirement: To pass the subject, students must obtain at least: 50% overall 30/70 in the written examination All Intended Learning Outcomes (ILOs) are addressed in all assessment components |
| **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:** | On completion of this subject students should have the following skills:<br># Application of knowledge of basic science and engineering fundamentals<br># Effective communication about computational efficiency<br># Capacity to reason and solve problems<br># Ability to undertake problem identification, formulation and solution<br># Capacity for creativity and innovation<br># Profound respect for truth and intellectual integrity, and for the ethics of scholarship |
| **Notes:** | **LEARNING AND TEACHING METHODS**<br>The subject involves weekly three-hour lectures. The lectures are a mix of direct delivery and interactive student problem solving. Although written assignments are submitted by students individually, in-plenum discussion of the problems is allowed, and encouraged.<br>**INDICATIVE KEY LEARNING RESOURCES**<br>Students are provided with lecture slides, and links on the LMS to the in-house animated software *Algorithms in Action*. The slides are integrated with the well-established textbook.<br>**CAREERS / INDUSTRY LINKS**<br>With Big Data at the forefront of modern computing solutions, industry is ever-more focused on efficient computational analysis methods. Software engineers, developers and data analysts will find not only the analysis techniques, but also the fundamental algorithmic design concepts, highly applicable to the handling of significant datasets. Building on an initial connection in a similar undergraduate offering, there is scope for industry liaison with this subject. |
| **Related Course(s):** | Master of Engineering in Distributed Computing<br>Master of Information Technology<br>Master of Information Technology<br>Master of Information Technology<br>Master of Operations Research and Management Science<br>Master of Philosophy - Engineering<br>Master of Science (Bioinformatics)<br>Ph.D.- Engineering |
| **Related Majors/Minors/ Specialisations:** | Master of Engineering (Mechatronics)<br>Master of Engineering (Software with Business)<br>Master of Engineering (Software) |