

433-633 Constraint Programming

Credit Points:	12.50
Level:	9 (Graduate/Postgraduate)
Dates & Locations:	This subject is not offered in 2009.
Time Commitment:	Contact Hours: 24 hours of lectures, 11 hours of workshops; Non-contact time commitment: 84 hours Total Time Commitment: Not available
Prerequisites:	Mathematical maturity and 4 computer science subjects at the level 3.
Corequisites:	None
Recommended Background Knowledge:	None
Non Allowed Subjects:	None
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>
Subject Overview:	<p>Constraint Programming is used to solve constraint satisfaction problems such as scheduling and allocation, which are of vital importance to modern business. Increasingly the discipline is replacing operations research, as a generic approach to solving management decision problems.</p> <p>Topics covered include: Constraints: valuations, modelling, constraint satisfaction, other constraint domains, Gaussian elimination, Simplex, other constraint solvers. Constraint simplification, projection and optimization. Finite constraint domains: constraint satisfaction problems, backtracking solvers, node and arc consistency, bounds consistency, generalised consistency methods. Constraint logic programs: user-defined constraints, programming with rules, evaluation, derivation trees, the CLP scheme. Simple modelling: choice, iteration, optimisation. Using data structures: records, lists, binary trees, hierarchical modelling. Controlling search: rule ordering, literal ordering, adding redundant constraints, minimisation. Programming with finite domain constraints: domains, labelling, different problem models. Advanced programming techniques: extending the solver, combined symbolic and arithmetic reasoning, programming optimisation, negation, dynamic scheduling.</p>
Objectives:	-
Assessment:	Project work of approximately 48-hours undertaken during semester (30%) and one 3-hour written examination at the end of semester (70%).
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 successful completion, students should:</p> <ul style="list-style-type: none"> # be able to build constraint programming solutions to combinatorial problems of moderate complexity; # have improved understanding of constraint solving algorithms; # be able to reason about the execution behaviour of constraint programs; # be able to undertake problem identification, formulation and solution;

	# have a capacity for independent critical thought, rational inquiry and self-directed learning; and # have a profound respect for truth and intellectual integrity, and for the ethics of scholarship.
Notes:	Credit may not be gained for both 433-433: Constraint Programming and 433-633: Constraint Programming.
Related Course(s):	Master of Software Systems Engineering