433-432 Logic Programming

Credit Points:	12.50
Level:	4 (Undergraduate)
Time Commitment:	Contact Hours: Twenty-four hours of lectures, 11 hours of workshops Total Time Commitment: Not available
Prerequisites:	Knowledge of predicate calculus and Prolog at the level taught in 433-255 Logic and Computation. Programming experience equivalent to the projects in 75 points of level 2 and 3 computer science subjects.
Corequisites:	None
Recommended Background Knowledge:	None
Non Allowed Subjects:	None
Core Participation Requirements:	<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>
Subject Overview:	The use of logic as a computational formalism originally grew out of natural language processing. Since then it has also been widely adopted in databases, rule-based systems, knowledge representation and formal methods. Logic programming languages offer a very high level approach to problem-solving and several novel programming techniques. Topics covered include: theoretical foundations: logical semantics, fixpoint semantics, SLD resolution; logic programming languages: logic variables, types and modes; logic programming techniques: structural induction, accumulators, difference lists, higher order programming, meta programming, program transformation; debugging: tracing, declarative debugging; implementation issues: indexing, tail recursion, management of backtracking.
Objectives:	-
Assessment:	Project work during semester, expected to take about 36 hours (40%) and a 3-hour end of semester written examination (60%). To pass the subject, students must obtain at least 50% overall, 20/40 on the projects and 30/60 in the examination.
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: # 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 433-432 Logic Programming and 433-632 Logic Programming.
Related Course(s):	Bachelor of Computer Science (Honours)

	Bachelor of Engineering (Computer Engineering) Bachelor of Engineering (Electrical Engineering) Bachelor of Engineering (Software Engineering)