433-632 Logic Programming

Credit Points:	12.50
Level:	9 (Graduate/Postgraduate)
Time Commitment:	Contact Hours: 24 hours of lectures, 11 hours of workshops; Non-contact time commitment: 84 hours Total Time Commitment: Not available
Prerequisites:	Knowledge of predicate calculus and Prolog, and general 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:	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
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. The working languages of the subject are Prolog and Mercury. The subject assumes some prior exposure to Prolog, but no exposure to Mercury.
Objectives:	-
Assessment:	Three projects of approx. 48 hours in total during semester (40%) and one 3-hour written examination at the end of the semester (60%). Both components must be completed satisfactorily to pass the subject.
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 successful completion, students should: # be able to use the programming techniques and tools appropriate to logic programming languages; # be familiar with the most important aspects of the technologies used to implement logic programming languages; # be able to explain the role of theory in the design of logic programming languages; # be able to explain the advantages of declarative programming languages over imperative programming languages; # 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):	Master of Software Systems Engineering