433-361 Programming Language Implementation

Credit Points:	12.500
Level:	Undergraduate
Dates & Locations:	2008, This subject commences in the following study period/s: Semester 2, - Taught on campus.
Time Commitment:	Contact Hours: Twenty-four hours of lectures and approximately 11 hours of tutorial classes Total Time Commitment: Not available
Prerequisites:	433-252 Software Engineering Principles and Tools, 433-253 Algorithms and Data Structures and 433-254 Software Design, one of which may be taken concurrently.
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 objectives of this subject are for students to be familiar with important concepts and techniques in programming language implementation; to develop a general appreciation of the constraints imposed on programming languages by implementation considerations; and to understand the basic ideas behind implementation of the programming language paradigms: imperative, object-oriented, functional and logic programming. Topics covered include compilers and interpreters: compiler structures; intermediate representations of programs; and global tables, run-time structures, lexical analysis, parsing, semantic analysis, code generation, optimisation.
Assessment:	A multi-stage project during the semester, expected to take about 36 hours (25%); and a 3-hour end-of-semester written examination (75%). To pass the subject, students must obtain at least 50% overall, 12.5/25 in project work and 37.5/75 in the written 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:	# ability to apply knowledge of basic science and engineering fundamentals # in-depth technical competence in at least one engineering discipline # ability to undertake problem identification, formulation and solution
Notes:	Students enrolled in the BSc (pre-2008 BSc), BASc or a combined BSc course will receive science credit for the completion of this subject.
Related Course(s):	Bachelor of Arts Bachelor of Arts and Bachelor of Science

	Bachelor of Arts and Sciences Bachelor of Engineering (Computer Engineering) Bachelor of Engineering (Electrical Engineering) Bachelor of Engineering (Mechatronics) and Bachelor of Computer Science Bachelor of Engineering (Software Engineering) Bachelor of Science