

433-431 Functional Programming

Credit Points:	12.50
Level:	4 (Undergraduate)
Dates & Locations:	This subject is not offered in 2009.
Time Commitment:	Contact Hours: Twenty-four hours of lectures, 11 hours of workshops Total Time Commitment: Not available
Prerequisites:	Knowledge of functional programming, and study at level 3 in at least four of the following areas: artificial intelligence, computer design, database systems, graphics, interactive system design, networks and communications, operating systems, programming languages, software engineering, and theory of computation.
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>Functional programming is an appealing programming language paradigm that all programmers should be aware of. Functional programming languages are generally simple and powerful and enjoy clean semantics. To learn a functional language, there are few rules to remember, and even fewer exceptions.</p> <p>Topics covered include functional programming languages: pattern matching, list comprehension, higher-order functions, strict and lazy evaluation; functional programming techniques: accumulators, higher-order programming, continuations, monads; functional data structures and algorithms; declarative input/output; types: polymorphism, type classes; computational models: evaluation of functional programs, lambda calculus, combinators, and graph reduction; implementation: translation to lambda expression and combinator form, abstract machines, strictness analysis, parallel evaluation.</p>
Objectives:	<p>On completion, students should be able to:</p> <ul style="list-style-type: none"> # appreciate the role of machine learning in AI theory and applications; # explain the main current techniques used to implement machine learning (data mining); # design and implement simple machine learning systems
Assessment:	Project work during semester, expected to take about 36 hours (40%) and a 2-hour end of semester written examination (60%).
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 completion of this subject students should:</p> <ul style="list-style-type: none"> # be able to undertake problem identification, formulation and solution;

	<ul style="list-style-type: none"># 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 433-431 Functional Programming and 433-631 Functional Programming.
Related Course(s):	Bachelor of Computer Science (Honours) Bachelor of Engineering (Computer Engineering) Bachelor of Engineering (Electrical Engineering) Bachelor of Engineering (Software Engineering)