

433-630 Principles of Programming Languages

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:	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>We communicate with computers in a variety of ways, but sophisticated requests need powerful formalisms for their expression. Programming languages form a surprisingly varied collection of notation for the specification of computing tasks. The study of programming languages is fascinating and important, as good programming language design and implementation is the enabling technology that allows us to improve software productivity and quality.</p> <p>Topics covered include: Programming language concepts: values, variables, bindings, types, execution and control, exceptions, nondeterminism, polymorphism, higher-order programming, encapsulation, abstract datatypes, modularity. Programming language paradigms: object-oriented, functional, and logic programming. Programming language design: syntax, semantics, abstraction, binding time, memory management, efficiency.</p>
Objectives:	-
Assessment:	Written essay of no more than 10 pages (20%), project work of approximately 12- hours during semester (5%), oral presentation of 15-minutes (5%) and one 2-hour written examination (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 program small applications in an object-oriented, a functional and a logic programming language; # be able to explain the differences, similarities and relative advantages of these three paradigms; # be able to explain abstract concepts from the theory of 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 433-430: Principles of Programming Languages and 433-630: Principles of Programming Languages
Related Course(s):	Master of Software Systems Engineering