COMP90045 Programming Language Implementation

Credit Points:	12.50
Level:	9 (Graduate/Postgraduate)
Dates & Locations:	2012, Parkville This subject commences in the following study period/s: Semester 1, Parkville - Taught on campus.
Time Commitment:	Contact Hours: 36 hours, made up of 24 one-hour lectures (two per week) and 12 one-hour workshops (one per week) Total Time Commitment: 120 hours
Prerequisites:	One of the following:

Subject	Study Period Commencement:	Credit Points:
COMP90048 Declarative Programming	Semester 2	12.50
COMP30020 Declarative Programming	Semester 1	12.50

Corequisites:	None
Recommended Background Knowledge:	None
Non Allowed Subjects:	433-361 Programming Language Implementation
Core Participation Requirements:	For the purposes of considering request for Reasonable Adjustments under the Disability Standards for Education (Cwth 2005), and Students Experiencing Academic Disadvantage Policy, academic requirements for this subject are articulated in the Subject Description, Subject Objectives, Generic Skills and Assessment Requirements of this entry.The University is dedicated to provide support to those with special requirements. Further details on the Disability support scheme can be found at the Disability Liaison Unit Website:http://www.services.unimelb.edu.au/disability/
Coordinator:	Dr Zoltan Somogyi
Contact:	Dr Aaron Harwood email: **aharwood@unimelb.edu.au (mailto:aharwood@unimelb.edu.au)**
Subject Overview:	This subject describes how compilers analyse source programs, how they translate them to target programs, and what tools are available to support these two tasks (analysis and synthesis). Topics covered include compiler structures; lexical analysis; syntax analysis; semantic analysis; intermediate representations of programs; code generation; and optimisation.
Objectives:	On completion of this subject students should be able to: # Describe important concepts and techniques in programming language implementation # Exploit their knowledge of compilers to be more effective programmers # Use analysis tools to help implement programs whose input has a structure that is non-trivial to recognize # Use synthesis tools to help implement programs that generate commands for other programs

Assessment:	A multi-stage project during the semester, expected to take about 36 hours (30%) And a 3-hour end-of-semester written examination (70%) To pass the subject, students must obtain 15/30 in project work and 35/70 in the written examination.
Prescribed Texts:	TBA
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 the subject students should have the: # Ability to undertake problem identification, formulation, and solution # Ability to utilise a systems approach to complex problems and to design and operational performance # Ability to manage information and documentation # Capacity for creativity and innovation
Related Course(s):	Bachelor of Computer Science (Honours) Master of Engineering in Distributed Computing Master of Science (Computer Science) Master of Software Systems Engineering
Related Majors/Minors/ Specialisations:	B-ENG Software Engineering stream Computer Science Master of Engineering (Software)