COMP90053 Program Analysis and Transformation

Credit Points:	12.50
Level:	9 (Graduate/Postgraduate)
Dates & Locations:	This subject is not offered in 2012.
Time Commitment:	Contact Hours: 36 hours, made up of 24 one-hour lectures (two per week) and 12 one-hour workshops (one per week)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:
COMP30020 Declarative Programming	Semester 1	12.50
COMP90048 Declarative Programming	Semester 2	12.50

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 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/
Contact:	Associate Professor Tim Baldwin email: **tbaldwin@unimelb.edu.au (mailto:tbaldwin@unimelb.edu.au)**
Subject Overview:	Programmers analyse and transform programs every day, e.g. to look for security problems and to add runtime checks that fix them. Many such tasks can now be done automatically, making programmers more productive, and yielding better results (since computers don't get bored or make mistakes). This subject will introduce the main techniques and applications of program analysis and transformation. Topics will include the following: semantics of programming languages; program representations; principles of program analysis, including collecting semantics, abstract interpretation, constraint-based analysis, and using lattices in fixpoint iterations; and meta-interpreters and program transformations. Additional topics will be selected from the following: program analyses for advanced type systems; analysing programs for statically detectable violations of safety and/or security policies, and program transformations to detect such violations at runtime; analyses and transformations for finding and implementing parallelism; program analyses and transformations for program optimization; program transformations for various forms of profiling; techniques for analysing very large programs; and declarative debugging.
Objectives:	On completion if this subject students should be able to: # Describe the standard approaches to program analysis and program transformation # Analyse the needs of a programming project and select and apply any existing program analysis and/or transformation tool that can help in the development of that project # Adapt any relevant existing program analyses and transformations to the needs of a project

	# Explain the limits of program analysis as applied to specific languages, and use this to inform decisions about what languages to use in Programming projects
Assessment:	Project work during semester, expected to take about 36 hours (30%) Amid-semester test (10%) And a 2-hour end-of-semester written examination (60%) To pass the subject, students must obtain: 15/30 in project work And 35/70 in the mid-semester test and end-of-semester written examination combined
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 completion of this subject students should have the: # Ability to undertake problem identification, formulation and solution # Ability to utilise a systems approach to complex problems and todesign an operational performance # Capacity for creativity and innnovaiton
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 Master of Engineering (Software)