SWEN90007 Software Design and Architecture

Credit Points:	12.50
Level:	9 (Graduate/Postgraduate)
Dates & Locations:	This subject is not offered in 2013.
Time Commitment:	Contact Hours: 36 hours, comprising of two 1-hour lectures and one 1-hour workshop per week Total Time Commitment: 120 hours
Prerequisites:	<table><tr><th>Subject</th><th>Study Period Commencement:</th><th>Credit Points:</th></tr><tr><td>SWEN90006 Software Engineering Methods</td><td>Not offered 2013</td><td>12.50</td></tr><tr><td>SWEN90008 Software Processes and Management</td><td>Not offered 2013</td><td>12.50</td></tr></table> OR 433-429 Software Engineering Methods 433-428 Software Processes and Management
Corequisites:	None
Recommended Background Knowledge:	433-606 Modelling Complex Software Systems
Non Allowed Subjects:	Students cannot enrol in and gain credit for this subject and: <table><tr><th>Subject</th><th>Study Period Commencement:</th><th>Credit Points:</th></tr><tr><td>SWEN90002 Engineering for Internet Applications</td><td>Not offered 2013</td><td>12.50</td></tr></table>
Core Participation Requirements:	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. 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
Contact:	Email: **tmiller@unimelb.edu.au (mailto:tmiller@unimelb.edu.au)**
Subject Overview:	One of the main challenges in developing enterprise wide distributed systems is in choosing the right software architectures. In this subject students will study software architectures in depth and the principles, techniques and tools for creating, developing and evaluating software architectures.
Objectives:	On completion of this subject, students should be able to: # Analyse large scale and distributed systems and select appropriate architectures for them # Evaluate architectures both qualitatively and quantitatively # Make suitable trade-offs between different architectures
Assessment:	A multi-staged, pair-based project in which students propose features of a system with properties commonly found in large-scale distributed enterprise systems, engineer an architectural design for that systems, and implement and evaluate it using industry-standard

	enterprise tools and frameworks. The project is expected about 36 hours, and will consist of a three reports totalling about 2000 words (30%). The four stages of the project are due in weeks 3 (nominate features of the system), 7 (architectural design), 9 (re-factored design), and 12 (implementation and evaluation) respectively. A three-hour end-of-semester written examination (70%). To pass the subject, a student must obtain: at least 50% overall;at least 50% (15/30) in project work; andat least 50% (35/70) in the written examination. ILOs 1-3 will be addressed by both the pair project and the end-of-semester written exam.
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 apply knowledge of science and engineering fundamentals # Ability to undertake problem identification, formulation, and solution # Ability to utilise a systems approach to complex problems and to design and operational performance # Proficiency in engineering design # Ability to manage information and documentation # Capacity for creativity and innovation # Ability to function effectively as an individual and in multidisciplinary and multicultural teams, as a team leader or manager as well as an effective team member # Capacity for lifelong learning and professional development
Related Course(s):	Master of Philosophy - Engineering Ph.D.- Engineering
Related Majors/Minors/ Specialisations:	B-ENG Software Engineering stream Master of Engineering (Software)