COMP90048 Declarative Programming

<table>
<tr><td><b>Credit Points:</b></td><td>12.50</td></tr>
<tr><td><b>Level:</b></td><td>9 (Graduate/Postgraduate)</td></tr>
<tr><td><b>Dates &amp; Locations:</b></td><td>2012, Parkville
This subject commences in the following study period/s:
Semester 2, Parkville - Taught on campus.</td></tr>
<tr><td><b>Time Commitment:</b></td><td>Contact Hours: 24 one-hour lectures (two per week) and 12 one-hour workshops (one per week) Total Time Commitment: 120 hours</td></tr>
<tr><td><b>Prerequisites:</b></td><td>None</td></tr>
<tr><td><b>Corequisites:</b></td><td>None</td></tr>
<tr><td><b>Recommended Background Knowledge:</b></td><td>Object Oriented Programming</td></tr>
<tr><td><b>Non Allowed Subjects:</b></td><td><table>
<tr><th>Subject</th><th>Study Period Commencement:</th><th>Credit Points:</th></tr>
<tr><td>COMP30020 Declarative Programming</td><td>Semester 1</td><td>12.50</td></tr>
</table></td></tr>
<tr><td><b>Core Participation Requirements:</b></td><td>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/</td></tr>
<tr><td><b>Coordinator:</b></td><td>Dr Peter Schachte</td></tr>
<tr><td><b>Contact:</b></td><td>Dr Peter Schachte
email: <b>schachte@unimelb.edu.au (mailto:schachte@unimelb.edu.au)</b></td></tr>
<tr><td><b>Subject Overview:</b></td><td>Declarative programming languages provide elegant and powerful programming paradigms that every programmer should know. Topics covered include functional programming, logic programming, constraint programming; declarative programming techniques, including higher order programming and the exploitation of advanced type systems; declarative languages as a competitive advantage, and how they fit into an environment dominated by imperative languages.</td></tr>
<tr><td><b>Objectives:</b></td><td>On completion of this subject, students should be able to:
Apply declarative programming techniques
Write medium size programs in a declarative language
Write programs in which different components use different languages; and
Select appropriate languages for each component task in a project</td></tr>
<tr><td><b>Assessment:</b></td><td>Project work during semester, expected to take about 36 hours (40%) A mid-semester test (10%) And a 2-hour end-of-semester written examination (50%) To pass the subject, students must obtain at least: 50% overall 20/40 in project work And 30/60 in the mid-semester test and end-of-semester written examination combined.</td></tr>
<tr><td><b>Prescribed Texts:</b></td><td>None</td></tr>
</table>

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 developed the following generic skills: # Ability to undertake problem identification, formulation and solution # Ability to utilise a systems approach to design and operational performance # Intellectual curiosity and creativity, including understanding of the philosophical and methodological bases of research activity # Openess to new ideas and unconventional critiques of received wisdom # Capacity for independent critical thought, rational inquiry and self-directed learning.
Related Course(s):	Master of Engineering in Distributed Computing Master of Science (Computer Science) Master of Software Systems Engineering
Related Majors/Minors/ Specialisations:	Computer Science Master of Engineering (Software)