# COMP30020 Declarative Programming

| | |
|---|---|
| **Credit Points:** | 12.50 |
| **Level:** | 3 (Undergraduate) |
| **Dates & Locations:** | 2010, Parkville<br>This subject commences in the following study period/s:<br>Semester 1, Parkville - Taught on campus. |
| **Time Commitment:** | Contact Hours: 24 one-hour lectures (two per week) and 12 one-hour workshops (one per week) Total Time Commitment: 120 hours |
| **Prerequisites:** | The prerequisites are:<br>433-298 Algorithms and Data Structures OR 433-521 Algorithms and Complexity OR 433-253 Algorithms and Data Structures |
| **Corequisites:** | None |
| **Recommended Background Knowledge:** | Object Oriented Programming |
| **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/ |
| **Coordinator:** | Dr Lee Naish, Dr Zoltan Somogyi |
| **Contact:** | Melbourne School of Engineering Office<br>Building 173, Grattan Street<br>The University of Melbourne<br>VIC 3010 Australia<br>General telephone enquiries<br>+ 61 3 8344 6703<br>+ 61 3 8344 6507<br>Facsimiles<br>+ 61 3 9349 2182<br>+ 61 3 8344 7707<br>Email<br>**eng-info@unimelb.edu.au (mailto:eng-info@unimelb.edu.au)** |
| **Subject Overview:** | 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. |
| **Objectives:** | On completion of this subject, students should be able to:<br># Apply declarative programming techniques,<br># Write medium size programs in a declarative language,<br># Write programs in which different components use different languages; and<br># Select appropriate languages for each component task in a project. |
| **Assessment:** | Project work during semester, expected to take about 36 hours (30%); a mid-semester test (10%); and a 2-hour end-of-semester written examination (60%). To pass the subject, students |

| | |
|---|---|
| | must obtain at least 50% overall, 15/30 in project work, and 35/70 in the mid-semester test and end-of-semester written examination combined. |
| **Prescribed Texts:** | TBA |
| **Breadth Options:** | This subject potentially can be taken as a breadth subject component for the following courses:<br># **Bachelor of Arts (https://handbook.unimelb.edu.au/view/2010/B-ARTS)**<br># **Bachelor of Commerce (https://handbook.unimelb.edu.au/view/2010/B-COM)**<br># **Bachelor of Environments (https://handbook.unimelb.edu.au/view/2010/B-ENVS)**<br># **Bachelor of Music (https://handbook.unimelb.edu.au/view/2010/B-MUS)**<br>You should visit **learn more about breadth subjects (http://breadth.unimelb.edu.au/breadth/info/index.html)** and read the breadth requirements for your degree, and should discuss your choice with your student adviser, before deciding on your subjects. |
| **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:<br># Ability to undertake problem identification, formulation and solution;<br># Ability to utilise a systems approach to design and operational performance ;<br># Intellectual curiosity and creativity, including understanding of the philosophical and methodological bases of research activity;<br># Openness to new ideas and unconventional critiques of received wisdom;<br># Capacity for independent critical thought, rational inquiry and self-directed learning. |
| **Related Course(s):** | Bachelor of Engineering (Software Engineering)<br>Bachelor of Science |
| **Related Majors/Minors/ Specialisations:** | Computer Science<br>Computer Science<br>Master of Engineering (Software)<br>Software Systems |