COMP90046 Constraint Programming

Credit Points:	12.5
Level:	9 (Graduate/Postgraduate)
Dates & Locations:	2015, Parkville This subject commences in the following study period/s: Semester 2, Parkville - Taught on campus.
Time Commitment:	Contact Hours: 36 hours, comprising of two 1-hour lectures and one 1-hour workshop per week Total Time Commitment: 200 hours
Prerequisites:	**One of the following:**

Subject	Study Period Commencement:	Credit Points:
COMP20003 Algorithms and Data Structures	Semester 2	12.50
COMP20007 Design of Algorithms	Semester 1	12.50
COMP90038 Algorithms and Complexity	Semester 1, Semester 2	12.50
COMP20006 Programming the Machine	Not offered 2015	12.50

Corequisites:	None
Recommended Background Knowledge:	None
Non Allowed Subjects:	433-433 Constraint Programming 433-633 Constraint Programming 433-671 Constraint Programming
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
Coordinator:	Prof Peter Stuckey
Contact:	email: **pstuckey@unimelb.edu.au (https://mce_host/faces/htdocs/@unimelb.edu.au)**
Subject Overview:	**AIMS** The aims for this subject is for students to develop an understanding of approaches to solving combinatorial optimization problems with computers, and to be able to demonstrate proficiency in modelling and solving programs using a high-level modelling language, and understanding of different solving technologies. The modelling language used is MiniZinc. **INDICATIVE CONTENT** Topics covered will include: # Modelling with constraints

	# Algorithms for manipulating constraints # Finite domain constraint solving # Global constraints # Linear Programming # Network Flow # Mixed Integer Programming # Boolean Satisfiability # Hybrid constraint solving.
Learning Outcomes:	**INTENDED LEARNING OUTCOMES (ILO)** On completion of this subject the student is expected to: 1 Model a complex constraint problem using a high level modelling language 2 Define and explore different search strategies for solving a problem 3 Explain how modelling interacts with solving algorithms, and formulate models to take advantage of this using state of the art optimisation tools 4 Explain different optimization technologies, and their strengths and weaknesses
Assessment:	Project assignments will be done during the semester, requiring approximately 35 – 40 hours in total (30%). There are three projects. The first one is due approximately in the third week and is worth 8% marks, requiring approximately 10 - 12 hours of work. The second project is due around mid-semester and is worth 10% marks, requiring approximately 13 - 15 hours of work. The third project is due at the end of the semester and is worth 12% marks, requiring approximately 15 - 18 hours of work One 3-hour end-of-semester examination (70%). Hurdle requirement: To pass the subject, students must obtain at least: 50% overall 15/30 in the project assignments 35/70 in the end-of-semester written examination. Intended Learning Outcomes (ILOs) 1, 2, 3, and 4 are addressed in the lectures, laboratory exercises, project assignments and the end-of-semester examination
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 be able to have the following skills: # Undertake problem identification, formulation, and solution # Utilise a systems approach to complex problems and to design and for operational performance # Manage information and documentation in solution creation # Demonstrate improved capacity for creativity and innovation.
Notes:	**LEARNING AND TEACHING METHODS** The subject comprises a weekly 2 hour lecture followed by a 1 hour laboratory exercise. Weekly readings are assigned from the textbook, and laboratory exercises are assigned. Additionally, a significant amount of project work is assigned. **INDICATIVE KEY LEARNING RESOURCES** At the beginning of the year, the coordinator will propose a textbook on constraint programming and will be made available through University Book Shop and library. The current suggested textbook is Programming with Constraints: an Introduction. Kim Marriott and Peter J. Stuckey, MIT Press. 1998. **CAREERS / INDUSTRY LINKS** The IT industry is a large and steadily growing industry. Increasingly companies are seeking to use optimization technology to provide decision support, assist in strategic and tactical planning, and manage daily operations. Modelling skills and understanding of optimization technology

	are essential for working in the optimization industry, for example in optimization consulting companies, or within the strategic planning groups within any major company. Most large companies have many problems that require optimization technology to be solved. Modelling and solving skills are also vital for employees whose role is to tackle these problems.
Related Course(s):	Master of Information Technology Master of Information Technology Master of Philosophy - Engineering Master of Science (Computer Science) Ph.D.- Engineering
Related Majors/Minors/ Specialisations:	Approved Masters level subjects from other departments B-ENG Software Engineering stream Computer Science Computer Science MIT Computing Specialisation MIT Distributed Computing Specialisation Master of Engineering (Software)