



**MONASH** University  
Information Technology

**FIT1029**  
**Algorithmic problem solving**

**Unit Guide**

**Semester 2, 2012**

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

*Last updated: 11 Jul 2012*

# Table of Contents

<b><u>FIT1029 Algorithmic problem solving - Semester 2, 2012</u></b> .....	<b>1</b>
<u>Mode of Delivery</u> .....	1
<u>Contact Hours</u> .....	1
<u>Workload</u> .....	1
<u>Chief Examiner</u> .....	1
<u>Campus Lecturer</u> .....	1
<u>Clayton</u> .....	1
<b><u>Academic Overview</u></b> .....	<b>2</b>
<u>Outcomes</u> .....	2
<u>Graduate Attributes</u> .....	2
<u>Assessment Summary</u> .....	3
<u>Teaching Approach</u> .....	3
<u>Feedback</u> .....	3
<u>Our feedback to You</u> .....	3
<u>Your feedback to Us</u> .....	3
<u>Previous Student Evaluations of this unit</u> .....	4
<u>Recommended Resources</u> .....	4
<b><u>Unit Schedule</u></b> .....	<b>5</b>
<b><u>Assessment Requirements</u></b> .....	<b>6</b>
<u>Assessment Policy</u> .....	6
<u>Assessment Tasks</u> .....	6
<u>Participation</u> .....	6
<u>Examinations</u> .....	7
<u>Examination 1</u> .....	7
<u>Assignment submission</u> .....	7
<u>Online submission</u> .....	7
<u>Extensions and penalties</u> .....	7
<u>Returning assignments</u> .....	8
<b><u>Other Information</u></b> .....	<b>9</b>
<u>Policies</u> .....	9
<u>Student services</u> .....	9
<u>Reading list</u> .....	10

# **FIT1029 Algorithmic problem solving - Semester 2, 2012**

Algorithms are recipes for solving a problem. They are fundamental to computer science and software engineering. Algorithms are the formal foundation of computer programming but also exist independently of computers as systematic problem-solving procedures. This unit introduces algorithmics, the study of algorithms. It is not about programming and coding but rather about understanding and analysing algorithms and about algorithmic problem-solving, i.e. the design of systematic problem-solving procedures. The unit will not require any knowledge of a programming language and is very hands-on. Students will develop algorithms to solve a wide variety of different problems, working individually as well as together in groups and as a class.

Topics include: what is a computational problem and what is an algorithm; basic control structures; basic data structures; modular algorithm structure; recursion; problem-solving strategies for algorithm development; arguing correctness of an algorithm; arguing termination of an algorithm; understanding the efficiency of an algorithm; and limitations of algorithms.

## **Mode of Delivery**

Clayton (Day)

## **Contact Hours**

2 hrs lectures/wk, 2 hrs tutorials/wk

## **Workload**

Students will be expected to spend 12 hours per week on various activities including:

- two-hour lecture and
- two-hour tutorial (requiring advance preparation)
- a minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.
- You will need to allocate up to 2 hours per week in some weeks, for use of a computer, including time for newsgroups/discussion groups.

## **Chief Examiner**

Dr David Albrecht

## **Campus Lecturer**

## **Clayton**

**Dr. Peter Tischer**

# Academic Overview

## Outcomes

At the completion of this unit students will have -  
A knowledge and understanding of:

- the difference between algorithms and processes;
- basic ways to structure algorithms: basic data structures (simple variables, collections structure, specifically vectors, lists, sets, and tables); basic control structures (sequence, choice, iteration);
- recursion;
- modular algorithm structures;
- the equivalence of recursion and iteration;
- problem solving strategies suitable for algorithm development including top-down design and bottom-up design;
- simple standard patterns for algorithms (eg traversal, search);
- what makes a good algorithm
- limitations of algorithms (high level).

Developed the skills to:

- develop simple iterative and recursive algorithms
- argue the correctness of simple algorithms
- judge the efficiency of simple algorithms, and

Developed attitudes that enable them to:

- value clear specification of problems;
- understand the relation between algorithms and programs;
- appreciate the value of designing abstract algorithms before starting to code a program;
- have confidence that they can develop algorithms to solve computational problems;
- appreciate that seemingly difficult problems can have very simple elegant algorithmic solutions (and vice versa);
- value correctness arguments for algorithms; and
- value the importance of simplicity and efficiency.

Demonstrated the communication skills necessary to:

- solve a problem by discussing possible approaches and solutions as a team; and
- clearly communicate (the specification of) a computational problem, its algorithmic solution and arguments for correctness and efficiency.

## Graduate Attributes

Monash prepares its graduates to be:

1. responsible and effective global citizens who:
  - a. engage in an internationalised world
  - b. exhibit cross-cultural competence
  - c. demonstrate ethical values

critical and creative scholars who:

- a. produce innovative solutions to problems
- b. apply research skills to a range of challenges
- c. communicate perceptively and effectively

## Assessment Summary

Examination (3 hours): 60%; In-semester assessment: 40%

Assessment Task	Value	Due Date
Assignment 1	10%	Monday 6 August 2012
Assignment 2	15%	Monday 3 September 2012
Assignment 3	15%	Monday, 1 October 2012
Examination 1	60%	To be advised

## Teaching Approach

### Lecture and tutorials or problem classes

This teaching and learning approach provides facilitated learning, practical exploration and peer learning.

## Feedback

### Our feedback to You

Types of feedback you can expect to receive in this unit are:

- Informal feedback on progress in labs/tutes
- Graded assignments with comments
- Test results and feedback

### Your feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through SETU, Student Evaluation of Teacher and Unit. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, and on student evaluations, see:

<http://www.monash.edu.au/about/monash-directions/directions.html>

<http://www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html>

## Previous Student Evaluations of this unit

This unit was first offered in 2010 and in semester 1, 2012, some of those students were taking third year units. Evidence gathered over the years indicates that the introduction of FIT1029 has helped to improve the students' programming skills and improved the pass rates in core BCS and BSE units. Feedback from students who have taken FIT1029 and has gone on to later year studies in the BCS and the BSE indicates that students feel that FIT1029 has helped them gain a better appreciation of algorithms and to regard thinking about algorithms as an activity that should be kept separated from coding algorithms into any particular computer programming language.

If you wish to view how previous students rated this unit, please go to <https://emuapps.monash.edu.au/unitevaluations/index.jsp>

## Recommended Resources

Recommended Texts are:

1. Levitin, A., *Introduction to the Design and Analysis of Algorithms (2nd Edition)*, Addison-Wesley, 2006
2. Harel, D. with Y. Feldman, *Algorithmics: The Spirit of Computing*, 3rd ed., Pearson Education Limited, 2004.

## Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	Introduction to the unit and the type of problems	
2	Understanding and modelling the problem	
3	Invariants in problems and data	Assignment 1 due Monday 6 August, 2012
4	Decomposition of problems and applying Brute Force to solve problems	
5	Using abstraction, symmetry, heuristics and divide and conquer to simplify problems	
6	Recursion	
7	Backtracking	Assignment 2 due Monday 3 September
8	Dynamic Programming	
9	Fundamentals	
10	Abstract Data Types and Correctness	Assignment 3 due Monday 1 October 2012
11	Complexity	
12	Limitations of algorithms	
	SWOT VAC	No formal assessment is undertaken SWOT VAC
	Examination period	LINK to Assessment Policy: <a href="http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html">http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html</a>

\*Unit Schedule details will be maintained and communicated to you via your MUSO (Blackboard or Moodle) learning system.

# Assessment Requirements

## Assessment Policy

Faculty Policy - Unit Assessment Hurdles

(<http://www.infotech.monash.edu.au/resources/staff/edgov/policies/assessment-examinations/unit-assessment-hu>)

Academic Integrity - Please see the Demystifying Citing and Referencing tutorial at

<http://lib.monash.edu/tutorials/citing/>

## Assessment Tasks

### Participation

#### • Assessment task 1

**Title:**

Assignment 1

**Description:**

This assignment will aim to help you understand how to go about finding algorithms to solve problems.

**Weighting:**

10%

**Criteria for assessment:**

Detailed assessment criteria will be issued along with the assignment.

1. All assumptions should be stated.
2. All algorithms must meet the problem specification.
3. Students should be able to answer questions about their own work.

**Due date:**

Monday 6 August 2012

#### • Assessment task 2

**Title:**

Assignment 2

**Description:**

This assignment will help you understand the importance of fundamental concepts, such as invariants, divide-and-conquer and induction, in developing algorithms.

**Weighting:**

15%

**Criteria for assessment:**

Detailed assessment criteria will be issued along with the assignment.

1. All assumptions should be stated.
2. All algorithms must meet the problem specification.
3. Students should be able to answer questions about their own work.

**Due date:**

Monday 3 September 2012



• **Assessment task 3**

**Title:**

Assignment 3

**Description:**

This assignment will help you understand different search techniques. It will also help you to communicate and reason about algorithms.

**Weighting:**

15%

**Criteria for assessment:**

Detailed assessment criteria will be issued along with the assignment.

1. All assumptions should be stated.
2. All algorithms must meet the problem specification.
3. Students should be able to answer questions about their own work.

**Due date:**

Monday, 1 October 2012

## Examinations

• **Examination 1**

**Weighting:**

60%

**Length:**

3 hours

**Type (open/closed book):**

Closed book

**Electronic devices allowed in the exam:**

None

## Assignment submission

It is a University requirement

(<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html>) for students to submit an assignment coversheet for each assessment item. Faculty Assignment coversheets can be found at <http://www.infotech.monash.edu.au/resources/student/forms/>. Please check with your Lecturer on the submission method for your assignment coversheet (e.g. attach a file to the online assignment submission, hand-in a hard copy, or use an online quiz).

## Online submission

If Electronic Submission has been approved for your unit, please submit your work via the VLE site for this unit, which you can access via links in the my.monash portal.

## Extensions and penalties

Submission must be made by the due date otherwise penalties will be enforced.

## Assessment Requirements

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process:

<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>.

## **Returning assignments**

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later.

## Other Information

### Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at:

<http://policy.monash.edu.au/policy-bank/academic/education/index.html>

Key educational policies include:

- Plagiarism  
(<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html>)
- Assessment  
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-p>)
- Special Consideration  
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.h>)
- Grading Scale  
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html>)
- Discipline: Student Policy  
(<http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html>)
- Academic Calendar and Semesters (<http://www.monash.edu.au/students/key-dates/>);
- Orientation and Transition (<http://www.infotech.monash.edu.au/resources/student/orientation/>);  
and
- Academic and Administrative Complaints and Grievances Policy  
(<http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy>)
- Codes of Practice for Teaching and Learning  
(<http://www.policy.monash.edu.au/policy-bank/academic/education/conduct/suppdocs/code-of-practice-tea>)

### Student services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at [www.monash.edu.au/students](http://www.monash.edu.au/students). For Sunway see <http://www.monash.edu.my/Student-services>, and for South Africa see <http://www.monash.ac.za/current/>

The Monash University Library provides a range of services and resources that enable you to save time and be more effective in your learning and research. Go to <http://www.lib.monash.edu.au> or the library tab in my.monash portal for more information. At Sunway, visit the Library and Learning Commons at <http://www.lib.monash.edu.my/>. At South Africa visit <http://www.lib.monash.ac.za/>.

Academic support services may be available for students who have a disability or medical condition. Registration with the Disability Liaison Unit is required. Further information is available as follows:

- Website: <http://monash.edu/equity-diversity/disability/index.html>;
- Email: [dlu@monash.edu](mailto:dlu@monash.edu)
- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus, or Student Community Services Department, Level 2, Building 2, Monash University, Sunway Campus
- Telephone: 03 9905 5704, or contact the Student Advisor, Student Community Services at 03 55146018 at Sunway

## Reading list

1. Levitin, A., *Introduction to the Design and Analysis of Algorithms (2nd Edition)*, Addison-Wesley, 2006
2. Harel, D. with Y. Feldman, *Algorithmics: The Spirit of Computing*, 3rd ed., Pearson Education Limited, 2004.
3. Michalewicz, Z. and M. Michalewicz, *Puzzle-Based Learning: An introduction to critical thinking, mathematics, and problem solving*, Hybrid Publishers, 2008.
4. Polya, G., *How to solve it; a new aspect of mathematical method*, 2nd ed., Garden City, N.Y., Doubleday, 1957
5. Bentley, J., *Programming Pearls*, Addison-Wesley, 1986
6. Bentley, J., *More Programming Pearls: confessions of a coder*, Addison-Wesley, 1988
7. Skiena, S., *The Algorithm Design Manual*, TELOS--the Electronic Library of Science, 1998
8. Cormen, T., C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 1990