

2501ICT

Programming 3

Semester 1 2010

Academic Organisation:	School of Information and Communication Technology
Faculty:	Science, Environment, Engineering and Technology
Credit point value:	10
Student Contribution Band:	Band 2
Course level:	Undergraduate
Campus/Location/Learning Mode:	Gold Coast / On Campus / In Person Nathan / On Campus / In Person
Convenor/s:	Dr Wayne Pullan (Gold Coast) Dr Rene Hexel (Nathan)
Enrolment Restrictions:	Restricted: Approval from Head of School
This document was last updated:	21 January 2010

BRIEF COURSE DESCRIPTION

This course extends students' programming skills through the introduction of alternative programming languages, the study of data structures, algorithms and concurrency, and the use of software development tools.

Prerequisite 1005ICT

Click on the link provided for "Head of School Approval" for summer semester enrolment.

SECTION A – TEACHING, LEARNING AND ASSESSMENT

COURSE AIMS

The organisation of data extends across a number of information technology disciplines, including programming principles, and database and file system design and implementation. As such this course extends the introduction to programming providing advanced support for these programming-related courses in the Bachelor of Information Technology.

The study of data structures and associated algorithms has a crucial role to play in an information technology course. The correct organisation of data is essential for its efficient storage, access and retrieval, both for single programming tasks as well as for composite system requirements.

LEARNING OUTCOMES

Upon successful completion of this course, students should be able to:

- Demonstrate an appreciation of data as a resource and critically evaluate issues in its organisation and management.
- Understand the various data structures and algorithms, which are the basic building blocks used to develop appropriate storage structures and file organisations.
- Understand the principles behind the use of the abstract data types (ADT), which involves the separation of the issue of specification from implementation.
- Use primitive data types and constructs and commonly used data types and complex data structures.
- Understanding of more advanced program design and implementation techniques, problem solving paradigms and their application in more complex programming tasks.

CONTENT, ORGANISATION AND TEACHING STRATEGIES

Type	Hrs / Week	Weeks	Hours
Type	Hrs/Week	Weeks	Hours
Lectures	2	1-13	26
Tutorials	2	1-13	26
Private Study and Preparation	7 (average)	1-13	<u>91</u>
			143

- The course has four hours of class contact per week, with two hours of lectures and two hours of tutorial/laboratory work. Lectures cover the basic fundamentals and principles of the course topics, while the tutorial/laboratory sessions serve as problem solving areas, with specified problems being set for each week's work.
- The tutorial program will be providing exercises specifically designed to help with the assignments. Attendance and participation at tutorials is strongly recommended and expected.
- The lecture program will be supported by lecture notes published on the course web site. Attendance at lectures is strongly encouraged. Important course announcements may be made during the lectures and extra course content not contained in either this outline or lecture notes may also be presented.
- Students are encouraged to research and read the references given below and other material relevant to the course. The lecture notes are regarded as only guidelines and summaries to provide the basis for further reading. Other reference and support material will be listed on the course web-site and students are required to use their own resources to further develop their knowledge and skills.

CONTENT SUMMARY

This course builds on the concepts of programming, data structures, and algorithms introduced in 1001ICT Programming I and 1005ICT Programming II, broadening the theoretical knowledge and practical programming experience through the introduction of a programming language different from Programming I and II. Course content covers a selection of topics from the following list:

Topic	Lecture Content
1.	Introduction of a new Programming Language
2.	Object Oriented Design

3.	Pointers and References
4.	Standard APIs
5.	Linked Lists
6.	Recursion
7.	Stacks
8.	Queues
9.	Trees
10.	Graphs
11.	Prefix-, Infix-, and Postfix (RPN) Expressions
12.	Search Algorithms
13.	Sorting Algorithms
14.	Complexity Analysis
15.	Processes and Threads
16.	Synchronisation
17.	Local and Remote Inter-Process Communication

Reading material and references are provided on the course web site, in the corresponding lectures, and in the lecture notes.

ASSESSMENT

Item	Assessment Task	Weighting	Due Date and Time
1.	Assignment 1	20 %	appx. Week 6
2.	Assignment 2	30 %	appx. Week 11
3.	Final Examination (Closed book - 180min)	50 %	Exam Weeks

- Assignment 1 requires students to complete all tutorials exercises up to week 5. An additional 16-24 hours (appx.) are required to produce the final programs and documentation.
- Assignment 2 requires the completion of all tutorial exercises up to week 10. An additional 16-24 hours are required to produce the final programs and documentation.
- The final exam will be closed book and students will not be allowed any written material. Calculators will not be allowed.
- To be eligible to pass the course, students are required to demonstrate a reasonable degree of competence in the required course objectives as examined in each form of assessment.
- Non-compliance with any assignment submission criteria may result in the assignment not being marked.
- Assignment marks are not final until the assessment board meeting. Students whose assignment marks deviate substantially from the exam marks may need to sit an interview as part of the assessment for the corresponding assignment(s).

GRADUATE SKILLS

The assignments test the student's understanding of the various data structures and algorithms, their usage, and the ability to successfully design and implement solutions to problems requiring the use of efficient algorithms in a suitable language, using structured programming techniques.

The final exam assesses the ability of each student to individually demonstrate his / her competence in and mastery of the course's theoretical content.

Graduate Skills (select appropriate boxes)	Taught	Practised	Assessed
Effective communication (written, oral and interpersonal)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Information literacy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Problem solving	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Critical evaluation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Work autonomously / in teams	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Creativity and innovation	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ethical behaviour in social / professional / work environments	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Responsible, effective citizenship	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

TEACHING TEAM

Course Convenor

Convenor Details	Gold Coast
Campus Convenor	Dr Wayne Pullan
Email	w.pullan@griffith.edu.au
Office Location	G23.1.57
Phone	555 29002
Fax	55528066
Consultation times	To be advised.

Convenor Details	Nathan
Campus Convenor	Dr René Hexel
Email	r.hexel@griffith.edu.au
Office Location	N44 2.21
Phone	373 55041
Fax	373 55051
Consultation times	To be advised.

COURSE COMMUNICATIONS

Students can communicate with the Course Convenor and any other member of the teaching team in person during and after the scheduled class contact times. Students may also contact the course convenor and/or any other member of the teaching teams using either email, telephone or in the relevant

office during the nominated consulting times. Students can also communicate with the course convenor and each other about the content of the course or related topics using the Web Forum.

TEXTS AND SUPPORTING MATERIALS

Recommended references include:

- **C++ Programming - Program Design Including Data Structures**, D.S.Malik (2009), Thomson, ISBN 1-4239-0222-X
- **Programming in Objective-C 2.0**, Stephen Kochan (2009), Addison-Wesley, ISBN 0-321-56615-7
- **Objective-C Pocket Reference**, A. Duncan (2002), O'Reilly, ISBN 0-596-00423-0
- **Cocoa: Programming for OS X**, A. Hillegass, (2004), Addison Wesley, ISBN 0-321-21314-9
- **Data Structures and Algorithm in C++**, Michael T Goodrich et al, (2004), John Wiley & Sons, ISBN 0-471-42924-4
- **Data Structures and Program Design in C++**, Robert L Kruse & Alexander J Ryba, (1999), Prentice Hall, ISBN 0-13-768995-0
- **Data Structures and Algorithms in Java**, Adam Drozdek, USA 2001, Brooks/Cole, ISBN 0-534-37668-1
- **A Framework for Program Design and Data Structures**, 2nd edition, K. A. Lambert and M. Osborne, Brooks/Cole, Canada 2004, ISBN 0-534-339285-7

Other reference and support material will be listed on the course web site (accessible through Learning@Griffith).

SECTION B – ADDITIONAL COURSE INFORMATION

Any additional course material will be provided via the course web site (accessible through Learning@Griffith).

SECTION C – KEY UNIVERSITY INFORMATION

ACADEMIC MISCONDUCT

Students must conduct their studies at the University honestly, ethically and in accordance with accepted standards of academic conduct. Any form of academic conduct that is contrary to these standards is academic misconduct and is unacceptable.

Some students engage deliberately in academic misconduct, with intent to deceive. This conscious, pre-mediated form of cheating is one of the worst forms of fraudulent academic behaviour, for which the University has zero tolerance and for which penalties, including exclusion from the University, will be applied.

However the University recognises many students commit academic misconduct without intent to deceive. These students may be required to undertake additional educational activities to remediate their behaviour.

Specifically it is academic misconduct for a student to:

- **Cheat in examinations and tests** by communicating, or attempting to communicate, with a fellow individual who is neither an invigilator or member of staff; by copying, or attempting to copy from a fellow candidate; attempting to introduce or consult during the examination, any unauthorised printed or written material, or electronic calculating or information storage device; or mobile phones or other communication device, or impersonates another.
- **Fabricate results** by claiming to have carried out tests, experiments or observations that have not taken place or by presenting results not supported by the evidence with the object of obtaining an unfair advantage.
- **Misrepresent themselves** by presenting an untrue statement or not disclosing where there is a duty to disclose in order to create a false appearance or identity.
- **Plagiarise** by representing the work of another as their own original work, without appropriate acknowledgement of the author or the source. This category of cheating includes the following:
 1. collusion, where a piece of work prepared by a group is represented as if it were the student's own;
 1. acquiring or commissioning a piece of work, which is not his/her own and representing it as if it were, by
 - purchasing a paper from a commercial service, including internet sites, whether pre-written or specially prepared for the student concerned
 - submitting a paper written by another person, either by a fellow student or a person who is not a member of the University;
 2. duplication of the same or almost identical work for more than one assessment item;
 3. copying ideas, concepts, research data, images, sounds or text;
 4. paraphrasing a paper from a source text, whether in manuscript, printed or electronic form, without appropriate acknowledgement;
 5. cutting or pasting statements from multiple sources or piecing together work of others and representing them as original work;
 6. submitting, as one own work, all or part of another student's work, even with the student's knowledge or consent.

A student who willingly assists another student to plagiarise (for example by willingly giving them their own work to copy from) is also breaching academic integrity, and may be subject to disciplinary action.

Visit the following web sites for further details:

[Institutional Framework for Promoting Academic Integrity among Students](#)
[Academic integrity for students](#)

PLAGIARISM DETECTION SOFTWARE

The University uses plagiarism detection software. Students should be aware that your Course Convenor may use this software to check submitted assignments. If this is the case your Course Convenor will

provide more detailed information about how the detection software will be used for individual assessment items.

HEALTH AND SAFETY

Griffith University is committed to providing a safe work and study environment, however all students, staff and visitors have an obligation to ensure the safety of themselves and those whose safety may be affected by their actions. Staff in control of learning activities will ensure as far as reasonably practical, that those activities are safe and that all safety obligations are being met. Students are required to comply with all safety instructions and are requested to report safety concerns to the University.

General health and safety information can be obtained from http://www.griffith.edu.au/hrm/health_and_safety/

Information about Laboratory safety can be obtained from http://www.griffith.edu.au/ots/secure/health/content_labsafety.html

KEY STUDENT-RELATED POLICIES

All University policy documents are accessible to students via the University's Policy Library website at: www.griffith.edu.au/policylibrary. Links to key policy documents are included below for easy reference:

[Academic Calendar](#)

[Academic Standing, Progression and Exclusion Policy](#)

[Assessment Policy](#)

[Examinations Timetabling Policy and Procedures](#)

[Guideline on Student E-Mail](#)

[Health and Safety Policy](#)

[Institutional Framework for Promoting Academic Integrity Among Students](#)

[Policy on Student Grievances and Appeals](#)

[Student Administration Policy](#)

[Student Charter](#)

UNIVERSITY SUPPORT RESOURCES

The University provides many facilities and support services to assist students in their studies. Links to information about University support resources available to students are included below for easy reference:

[Learning Centres](#) - the University provides access to common use computing facilities for educational purposes. For details visit <https://intranet.secure.griffith.edu.au/computing/student-computing/finding-available-computers>

[Learning@Griffith](#) - there is a dedicated website for this course via the Learning@Griffith student portal.

[Student Services](#) facilitate student access to and success at their academic studies. Student Services includes: Careers and Employment Service; Chaplaincy; Counselling Service; Health Service; Student Equity Services (incorporating the Disabilities Service); and the Welfare Office.

[Learning Services](#) within the Division of Information Services provides learning support in three skill areas: computing skills; library skills; and academic skills. The study skills resources on the website include self-help tasks focusing on critical thinking, exam skills, note taking, preparing presentations, referencing, writing, proof reading, and time management.