The University of Adelaide  
Department of Computer Science  
Diploma in Computer Science 2000  
Certificate in Computer Science 2000

The Department of Computer Science offers programs leading to a Graduate Diploma and a Graduate Certificate in Computer Science for candidates with a Bachelor’s degree in a field other than Computer Science. The Diploma requires one year of full–time study, or the part-time equivalent, while the Certificate requires half a year of full-time study, or the part-time equivalent. The Diploma provides a means by which graduates may gain Computer Science knowledge equivalent to a Bachelor’s degree majoring in Computer Science.

The course has proved popular with candidates seeking to improve their prospects and to become well prepared for careers in Computer Programming and Systems Analysis in commerce, industry and the public service. The Diploma is accredited with the Australian Computer Society—upon completion of the course, students are eligible for membership of the society.

Entry Requirements
Entry into the programs requires a Bachelor’s degree in a field other than Computer Science. A quota restricts entry to the equivalent of 20 full–time students per year in each course. The criteria for selection are the applicant’s academic record and professional experience.

Course Content
The Diploma requires the study of a mixture of Level 2 and Level 3 Computer Science subjects. The mixture must total at least 24 points with the provision that at least 8 points of Level 2 and 10 points of Level 3 subjects must be presented. The Certificate requires the study of Level 2 (and possibly some Level 3) Computer Science subjects to at least 12 points value. Subjects available and the associated points values are summarised overleaf. You are, however, advised to check with the Department prior to selection of the course, students are eligible for membership of the society.

Entry Requirements
Entry into the programs requires a Bachelor’s degree in a field other than Computer Science. A quota restricts entry to the equivalent of 20 full–time students per year in each course. The criteria for selection are the applicant’s academic record and professional experience.

Course Content
The Diploma requires the study of a mixture of Level 2 and Level 3 Computer Science subjects. The mixture must total at least 24 points with the provision that at least 8 points of Level 2 and 10 points of Level 3 subjects must be presented. The Certificate requires the study of Level 2 (and possibly some Level 3) Computer Science subjects to at least 12 points value. Subjects available and the associated points values are summarised overleaf. You are, however, advised to check with the Department prior to selection of subjects in case some subjects are unavailable. Please note that not all Level 3 subjects can be taken if the Diploma is to be done in one year. This is due to pre–requisite requirements, and the set of available subjects varies (slightly) from year to year. There is only one compulsory course in the Diploma: Software Engineering and Project.

Although the courses assumes knowledge equivalent to first–year Computer Science, the department provides some assistance to students lacking this knowledge by a Level 2 course worth 3 credit points which, together with the preparatory reading program, covers the material presented in first year. This course runs for 4 weeks, commencing on January 24th 2000. This course is available to all students except those who have recently completed the department’s first-year course.

For any student having more than the minimum background, the course of study may be adjusted suitably. For students in this situation, your attention is drawn to the University’s Schedules for the “Master of Computer Science”. The course regulations permit subjects from those schedules to be submitted for the Diploma (and the Certificate). Students contemplating this action are required to consult with the coordinator to determine an appropriate course of study.

Arrangements for Part–Time Students
It is regretted that evening lectures are not available. Tutorials can be arranged for 5:15pm, and practical sessions can be scheduled in the evenings; the laboratories are open late and on weekends. Although it is not a formal requirement of University regulations, the department expects all students to attend lectures.

Lectures and Tutorials
Most of the courses studied will have two lectures per week for one semester. For Level 2 subjects there is typically a 1-hour tutorial every. Most subjects require practical work which takes about 2 hours per week (depending on ability). Some subjects (particularly those with “and Project” in the title) have large practical components and require more practical work.

Preparatory Reading
L. Goldschlager & A. Lister, Computer Science: A Modern Introduction (Prentice–Hall 1987)

Coordinator
Charles Lakos, Department of Computer Science, Room 1043, Phone: 8303 6172. Fax 8303 4366.
Admission and Enrolment
Apply through the South Australian Tertiary Admissions Centre (SATAC), 104 Frome Street, Adelaide, phone 8224 4000. If accepted, you will be informed in writing prior to enrolment in late January.

Fees
These courses are offered on a fee paying basis and fees may vary from year to year. However, the University offers a limited number of post-graduate scholarships that enable the cost to be deferred under the HECS. Scholarship selection is based on financial hardship and academic merit.

Australian residents should contact the Student Information Office (8303 5208) for further information. Overseas students should contact the Director of International Programs (8303 5251) for up-to-date information on requirements.

At the time of writing the fee for 2000 is expected to be $9,950 for 24 points. The fee is paid in instalments according to the number of points being studied, i.e., about $415 per point.

Quick Guide to Courses
You need to make up 24 points for the Diploma from the subjects shown in the following provisional time-table, with a minimum of 8 points from the Level 2 and 10 points from the Level 3 offerings. (The Certificate only requires 12 points, which may all be at the second year level.) At enrolment, the diploma coordinator will assist you in choosing a set of subjects appropriate to your interests and background. A summary of each subject is attached. Please note that, if necessary the stated pre-requisites for a course may be waived to suit individual study programs.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Semester</th>
<th>Points</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Level 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>¶ 9492 Computer Science Concepts</td>
<td>Summer</td>
<td>3</td>
<td>4pm</td>
<td>4pm</td>
<td>4pm</td>
<td>4pm</td>
<td>4pm</td>
</tr>
<tr>
<td>1956 Computer Systems</td>
<td>1</td>
<td>2</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>10am</td>
<td>12pm</td>
</tr>
<tr>
<td>5132 Data Structures &amp; Algorithms</td>
<td>1</td>
<td>2</td>
<td>–</td>
<td>10am</td>
<td>–</td>
<td>–</td>
<td>9am</td>
</tr>
<tr>
<td>3169 Database &amp; Information Systems</td>
<td>2</td>
<td>2</td>
<td>9am</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>12pm</td>
</tr>
<tr>
<td>3655 Numerical Methods</td>
<td>1</td>
<td>2</td>
<td>10am</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>10am</td>
</tr>
<tr>
<td>2430 Programming Paradigms</td>
<td>2</td>
<td>2</td>
<td>–</td>
<td>10am</td>
<td>–</td>
<td>10am</td>
<td>–</td>
</tr>
<tr>
<td>9877 Open Systems &amp; Client/Server Computing</td>
<td>2</td>
<td>2</td>
<td>10am</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>10am</td>
</tr>
<tr>
<td><strong>Level 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5141 Computer Architecture</td>
<td>1</td>
<td>2</td>
<td>3pm</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>3pm</td>
</tr>
<tr>
<td>* 1234 Compiler Construction &amp; Project</td>
<td>1</td>
<td>3</td>
<td>4pm</td>
<td>–</td>
<td>4pm</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>* 9820 Numerical Analysis</td>
<td>1</td>
<td>2</td>
<td>–</td>
<td>4pm</td>
<td>–</td>
<td>4pm</td>
<td>–</td>
</tr>
<tr>
<td>* 2382 Programming Techniques</td>
<td>1</td>
<td>2</td>
<td>–</td>
<td>2pm</td>
<td>–</td>
<td>2pm</td>
<td>–</td>
</tr>
<tr>
<td>6378 Artificial Intelligence</td>
<td>1</td>
<td>2</td>
<td>–</td>
<td>–</td>
<td>3pm</td>
<td>–</td>
<td>4pm</td>
</tr>
<tr>
<td>* 9811 Advanced Programming Paradigms</td>
<td>2</td>
<td>2</td>
<td>–</td>
<td>4pm</td>
<td>–</td>
<td>4pm</td>
<td>–</td>
</tr>
<tr>
<td>2328 Computer Networks &amp; Applications</td>
<td>2</td>
<td>2</td>
<td>3pm</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>3pm</td>
</tr>
<tr>
<td>4468 Operating Systems</td>
<td>2</td>
<td>2</td>
<td>–</td>
<td>2pm</td>
<td>–</td>
<td>2pm</td>
<td>–</td>
</tr>
<tr>
<td>‡ 7732 Systems Analysis &amp; Project</td>
<td>2</td>
<td>3</td>
<td>–</td>
<td>3pm</td>
<td>–</td>
<td>3pm</td>
<td>–</td>
</tr>
<tr>
<td>† 6263 Software Engineering &amp; Project</td>
<td>2</td>
<td>3</td>
<td>4pm</td>
<td>–</td>
<td>4pm</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

¶ This course is an intensive “crash course” given prior to the start of Semester 1, starting in the last week of January.
* These subjects may not normally be taken in a 1-year diploma due to pre-requisite constraints.
† All Diploma students are required to take 6263 Software Engineering and Project.
‡ Not available in 1999.
Level 2 Computer Science Subjects:

9492 Computer Science Concepts

This course is a 4-week intensive introduction to computer science. It is intended for students enrolling in the Graduate Diploma in Computer Science, except those who studied first-year Computer Science at the University of Adelaide.

Contact hours: The course format comprises two 50-minute lecture/tutorial units, each day from Monday to Thursday between 4 and 6pm. Consulting (one-on-one) is available at the same time on Fridays. The course involves assignment work which is done in our teaching laboratories, which are open until 10pm.

Content:
The focus of the course is to impart the basic programming skills necessary to undertake the Level 2 and Level 3 subjects that make up the rest of the diploma. The course has three basic components: algorithms & algorithm design (6 lectures), programming in Ada (12 lectures) and an introduction to computer systems (6 lectures). You will have to read the text and reference books as well as attend the lectures to successfully complete the course.

Assessment: An examination and some compulsory practical work.


1956 Computer Systems

Pre-requisite: 9492 Computer Science Concepts

Contact hours: 2 lectures and 2 hours of practical work a week, plus 1 tutorial a fortnight.

Content:
Machine data types and instructions, addressing modes, exceptions, implementing HLL semantics using the ISA, CPU organization, process management, principle of locality, main memory organization, cache, disk file systems, virtual memory, I/O controllers, DMA, I/O resource management, process memory buses, I/O buses and protocols, computer networks and protocols, network services and applications.

5132 Data Structures and Algorithms

Pre-requisite: 9492 Computer Science Concepts

Contact hours: 2 lectures and 2 hours of practical work a week, plus 1 tutorial a fortnight.

Content:
Records, sets, general files; program development techniques including basic ideas of correctness; stacks and queues; dynamic storage; pointers; linked lists; representation of stacks and queues, general list operations. Notions of complexity and analysis; notion of abstract data type; sets and sequences as examples; searching and information retrieval illustrated with a “table” abstract data type; various representations of a “table” abstract data type; recursion.

3655 Numerical Methods

Pre-requisite: 9492 Computer Science Concepts

Contact hours: 2 lectures and 2 hours of practical work a week, plus 1 tutorial a fortnight.

Content:

3169 Database and Information Systems

Pre-requisite: 9492 Computer Science Concepts

Contact hours: 2 lectures and 2 hours of practical work a week, plus 1 tutorial a fortnight.

Content:
The SQL database query language; queries, sub-queries, updates and transactions. Concurrency issues; record locking, deadlock and recovery. Distributed Databases. Database Security in SQL. The relational, hierarchical and network database models. Theory of relational databases; the relational algebra and relational calculus. Query transformation. Object-oriented databases. Practical use of a 4th generation system (e.g., Oracle) to generate screen-based forms, and reports. A low-level database programming language (e.g., Cobol or ESQL). Design of random access and sequential access update programs.

2430 Programming Paradigms

Pre-requisite: 5132 Data Structures and Algorithms.

Contact hours: 2 lectures and 2 hours of practical work a week, plus 1 tutorial a fortnight.

Content:
A study of four major programming approaches: imperative, functional, logic, and object-oriented. Imperative paradigms: procedural abstraction, parameter

9877 Open Systems and Client/Server Computing
Pre-requisite: 9492 Computer Science Concepts.
Contact hours: 2 lectures and 2 hours of practical work a week, plus 1 tutorial a fortnight.

Content:
Introduction to C programming, operating systems interfaces, Unix system services and libraries, X-windows, user interface programming, network services and interfaces, Internet protocols and programming, client/server model, client/server programming.

Level-3 Computer Science Subjects:
2328 Computer Networks and Applications
Contact hours: 2 lectures and 2 hours of practical work a week and 1 tutorial every 3 weeks.

Content:
Introduction to networks and digital communications: Nyquist and Shannon results, modulation and encoding techniques, transmission media and network topologies, switching techniques; The OSI reference model: functions and protocols of the seven layers; LANs, MANs and WANs: CSMA/CD, token bus, token ring, DQDB), FDDI and B-ISDN; Internetworking: bridges, routers, gateways, overview of TCP/IP, security.

1234 Compiler Construction and Project
Assumed knowledge: 2430 Programming Paradigms & 2382 Programming Techniques.
Contact hours: 2 lectures and 4 hours of practical work a week and 1 tutorial every 3 weeks.

Content:
The structure of compilers: lexical analysis, syntax analysis (top-down and bottom-up techniques), environmental handling, the handling of context-sensitive and context-free errors, type checking and code generation. Run-time support for Algol-like languages, including storage management. BNF languages and grammars. This course is closely coupled with the writing of a large, compulsory programming project.

9820 Numerical Analysis
Pre-requisite: 3655 Numerical Methods.
Contact hours: 2 lectures and 2 hours of practical work a week and 1 tutorial every 3 weeks.

Content:
Topics will include computer arithmetic, numerical solution of non-linear equations, numerical solution of systems of linear equations and the computation of eigenvalues and eigenvectors. The course is intended to be an analysis, rather than a methods course.

Equipment:
Pocket calculator with the elementary functions.

2382 Programming Techniques
Pre-requisite: 5132 Data Structures and Algorithms.
Contact hours: 2 lectures and 2 hours of practical work a week and 1 tutorial every 3 weeks.

Content:

6378 Artificial Intelligence
Pre-requisite: 5132 Data Structures and Algorithms.
Contact hours: 2 lectures and 2 hours of practical work a week and 1 tutorial every 3 weeks.

Content:
AI methodology and fundamentals; description matching and goal-reduction; ANALOGY; AND/OR trees; exploiting natural constraints: Waltz algorithm; search: hill-climbing, beam, best-first, A*; minimax procedure and alpha-beta pruning for game-playing; learning: parameter-adjustment and Winston near-miss/reinforcement procedure; means-end analysis and GPS; rule-based systems: forward- and backward-chaining, MYCIN, Xcon; generate and test paradigm with Dendral. Representation issues: inheritance, demons, defaults, perspectives, frames, primitives; aspects of Prolog; neural networks: recurrent backpropagation technique.

9811 Advanced Programming Paradigms
Pre-requisite: 5132 Data Structures and Algorithms.
Assumed knowledge: 2430 Programming Paradigms & 2382 Programming Techniques.
Contact hours: 2 lectures and 2 hours of practical work a week and 1 tutorial every 3 weeks.

Content:
Advanced functional programming in Miranda; the lambda-calculus and combinators; graph-reduction implementations; stream processing in Scheme (a dia-
lect of Lisp); lazy and strict evaluation; coroutines in functional and imperative paradigms. Parallel programming: shared memory process model; message passing; data parallel programming; parallel functional languages and implicit parallelism.

5141 Computer Architecture
**Pre-requisites:** 5132 Data Structures and Algorithms & 1956 Computer Systems.
**Contact hours:** 2 lectures and 2 hours of practical work a week and 1 tutorial every 3 weeks.
**Content:**
Fundamentals of computer design; quantifying cost and performance; instruction set architecture; program behaviour and measurement of instruction set use; processor datapaths control; pipelining, handling pipeline hazards; memory hierarchies and performance; I/O devices, controllers and drivers; I/O and system performance; multiprocessors and special purpose processors.

4468 Operating Systems
**Pre-requisites:** 5132 Data Structures and Algorithms & 1956 Computer Systems.
**Contact hours:** 2 lectures and 2 hours of practical work a week and 1 tutorial every 3 weeks.
**Content:**
OS purposes: resource management and the extended virtual computer; historical development. Processes: critical sections and mutual exclusion, semaphores, monitors, classical problems, deadlock; process scheduling. Input and Output: hardware and software control; disks, terminals, clocks. Memory management: multiprocessing needs; swapping; virtual memory, paging and segmentation; page replacement; File System: operations, implementation, performance, protection.

6263 Software Engineering and Project
**Pre-requisite:** 5132 Data Structures and Algorithms.
**Assumed knowledge:** 2382 Programming Techniques.
**Contact hours:** 2 lectures and 4 hours of practical work a week and 1 tutorial every 3 weeks.
**Content:**
This is a first course in software engineering and provides an introduction to the production of high quality software solutions to large tasks. Among the topics covered in this course are the following: models of the software life-cycle, requirements analysis and specification, program design techniques and paradigms, software specification techniques, configuration management and version control, quality assurance, integration and testing, project management, computer-aided software engineering and integrated software engineering environments.

7732 Systems Analysis and Project
**Pre-requisite:** 3169 Database & Information Systems.
**Contact hours:** 2 lectures and 4 hours of practical work a week and 1 tutorial every 3 weeks.
**Restriction:** Not to be taught in 1999.
**Content:**