



Curriculum  
Council

## **COMPUTER SCIENCE**

For teaching from 2010

## IMPORTANT INFORMATION

### Syllabus review

Once a course syllabus has been accredited by the Curriculum Council, the implementation of that syllabus will be monitored by the syllabus committee. This committee can advise council about any need for syllabus review. Syllabus change deemed to be minor requires schools to be notified of the change at least six months before implementation. Major syllabus change requires schools to be notified 18 months before implementation. Formal processes of syllabus review and requisite reaccreditation will apply.

### Other sources of information

The Western Australian Certificate of Education (WACE) Manual contains essential information on assessment, moderation and other procedures that need to be read in conjunction with this course.

The Curriculum Council will support teachers in delivering the course by providing resources and professional development online.

The council website [www.curriculum.wa.edu.au](http://www.curriculum.wa.edu.au) provides support materials including sample programs, assessment outlines, assessment tasks, with marking keys, sample examinations with marking keys and grade descriptors with annotated student work samples.

Training package support materials are developed by Registered Training Organisations (RTOs), government bodies and industry training advisory bodies to support the implementation of industry training packages. Approved support materials are listed at [www.ntis.gov.au](http://www.ntis.gov.au)

### WACE providers

Throughout this course booklet the term 'school' is intended to include both schools and other WACE providers.

### Copyright

© Curriculum Council, 2008.

This document—apart from any third party copyright material contained in it—may be freely copied or communicated for non-commercial purposes by educational institutions, provided that it is not changed in any way and that the Curriculum Council is acknowledged as the copyright owner.

Copying or communication for any other purpose can be done only within the terms of the Copyright Act or by permission of the Curriculum Council.

Copying or communication of any third party copyright material contained in this document can be done only within the terms of the Copyright Act or by permission of the copyright owners.

# Rationale

Information and communication technologies are integral to the 21st century global village and economy. Everyone is influenced by computers in some way. Whilst we all use computer systems as a means to an end, it is vital to develop an interest in the intricate workings of computer systems, so that future generations have the knowledge, understanding and skills to create and maintain computer systems. The Computer Science course aims to take students beyond the use of computers at an application level into the realm of creating software, building and networking computer-based systems.

Computer science is a wide-ranging discipline that leads to many different professional and non-professional careers. This course covers a variety of topics to appeal to a diverse range of students interested in academic or vocational pathways. This course aims to stimulate students' awareness of the nature and scope of computer science. It enables them to recognise the opportunities within this field for their own potential growth and the possibility of contributing to the development of our future technological society.

The course focuses on the fundamental principles and concepts within the field, as well as promoting flexibility and adaptability in the application of these principles to meet current computing trends. The underpinning knowledge and skills in computer science involve the principles related to the creation of computer systems, software and connectivity between computers used in the home, workplace and education. Students develop conceptual and technical skills as they learn how to diagnose and solve problems in the course of understanding the basic building blocks of computing.

Since the personal, social and professional lives of people are affected by technological developments and because innovation occurs at a rapid pace, it is imperative to consider the values dimension of any changes. Hence, a heightened awareness of the impact of technological developments on the personal, societal and professional lives of individuals and businesses is promoted, so that students may be assisted to make ethical decisions. The ethical, moral and legal constraints that have influenced or may govern any development will be presented whenever appropriate so that students recognise the consequences of decisions made in respect to the development and use of technology.

This course is designed to encourage students to study computer science as it is applied in the workforce and home or with the aim of pursuing further studies in the future. It gives students practical and interpersonal skills that equip them to function effectively in a world where these attributes are vital for employability and daily life in a technological society.

This course provides students with the opportunity to further their achievement of specific overarching learning outcomes from the Curriculum Framework together with the development of the core-shared values.

## Course outcomes

The Computer Science course is designed to facilitate the achievement of four outcomes. These outcomes are based on the Technology and Enterprise learning area outcomes in the Curriculum Framework. Outcomes are statements of what students should know, understand, value and be able to do as a result of the syllabus content taught.

### **Outcome 1: Technology process**

Students apply a technology process to develop computer-based systems.

In achieving this outcome, students:

- investigate ideas and generate proposals;
- develop solutions that meet specifications and recognised standards; and
- evaluate computer-based solutions.

### **Outcome 2: Knowledge and understanding of computer-based systems**

Students understand the design, application and interactions of hardware and software in computer-based systems.

In achieving this outcome, students:

- understand the appropriate selection and application of computer-based system components;
- understand the nature of the interactions between the elements of computer-based systems; and
- understand the concepts associated with computer-based systems.

### **Outcome 3: Skills for computer-based systems**

Students apply skills to maintain, adapt or develop computer-based systems.

In achieving this outcome, students:

- apply a range of problem-solving techniques when maintaining or developing computer-based systems;
- apply a range of conventions and standards when implementing a maintenance or development solution; and
- apply organisational skills to identify and use appropriate hardware and software resources when maintaining or developing a computer-based system.

#### **Outcome 4: Computer-based systems in society**

Students understand the interrelations between the development and use of computer-based systems, the individual and society.

In achieving this outcome, students:

- understand that developers' attitudes and values affect the development of computer-based systems;
- understand that users' attitudes and values affect the development and use of computer-based systems; and
- understand there are legal, societal and ethical impacts when computer-based systems are developed and adopted.

#### **Outcome progressions**

Each of the outcomes is described as a learning progression across six broad levels (see Appendix 1). In teaching a particular course unit, teachers can use the outcome progressions along with the unit content and contexts to:

- plan appropriate lessons and activities for their students, and
- develop specific assessment tasks and marking keys.

## **Course content**

The course content needs to be the focus of the learning program. It enables students to maximise their achievement of both the overarching learning outcomes from the Curriculum Framework and the Computer Science course outcomes.

The course content is divided into three areas:

- components
- design, development and management
- tools.

Typically about 60% of the time would be spent on the tools area and the rest of the time divided between components and design, development and management areas.

### **Components**

#### **Hardware components**

The course covers types of computers and their architecture. This includes computer hardware components (input, processing and output devices), storage devices and the internal components of a computer system (processor, mainboard, buses, cache and ports). The interrelationships and selection of hardware components in computer systems are addressed. Topics include computer organisation, including number systems, instruction sets, the execution cycle and the relationship between the central processing unit and memory.

The course covers the various structures and components of a network, including the communication media used to combine them. It looks at the convergence of technologies, which

involves the integration of computers and communication hardware and the design and creation of networks of various configurations, as well as connecting networks of different types together. The application of connectivity standards, relating to networks and the internet are addressed. Topics include communication software models and standards (e.g. ISO-OSI and TCP/IP models); the types, purpose and use of protocols, servers and operating systems in communication; and software and the aspects to consider in network security and organisation.

#### **Software components**

The course covers the types and purposes of software, the functionality of operating software in a computer system and the principles of systems software, such as structuring methods, abstractions, processes and resources. Topics include concepts of application program interfaces, data storage by operating systems and the techniques of concurrency, scheduling, dispatch, memory and device management used in operating systems. Different data and file types and the various methods used to access them are addressed.

### **Design, development and management**

Specifications about systems are detailed in the Specifications booklet.

#### **Systems**

The course covers hardware, communication, database management and software development systems. Design features of hardware systems and components as well as the factors affecting this design are considered. This is based on knowledge of the functions and technical capabilities of systems, how components are configured to form a computer system suitable for a particular context and factors which affect the design of a networked information system. The course considers the compatibility of components, their ability to produce required output, quality and information, and address specified criteria such as bandwidth, cost, ease of use, security, health and safety considerations. This extends to an evaluation of systems, devices or components and the knowledge and skills required when acquiring computer hardware, using various information techniques and sources.

A technology process is used in the design, modification and implementation of a computer system. The course addresses the management of a systems development project that may include some basic systems engineering and the application of standards. The course considers how a developer's interactions with a client affect the development and use of the system. Various methods of developing software systems, including top-down, bottom-up and prototyping and the

problems associated with connecting systems in an increasingly global environment are addressed.

### **Ethics, law and society**

The course considers ethical issues relating to data and information storage and the use of computer systems. Australian law and the various regulations concerning computer-based systems in Australia and internationally are covered. The course addresses the types of Australian and global industries that use computer systems and the factors that affect their use.

The course examines the issues relating to the storage and transmission of data in a computer system, including security, privacy, copyright, accountability and legal requirements of the systems developed. It considers the different perspectives a user and developer have to the development and use of computer-based systems. The course also considers the social implications and impacts on individuals, communities and environments, of using computer-based systems.

### **Tools**

Specifications about tools are detailed in the Specifications booklet.

### **Data representation and organisation**

The course addresses the distinction between data and information, including the different types of data (text, number etc.) and the varied representation of data within a computer (dates stored differently to text etc.). This also includes data structures used to represent data in different ways: using sequential files, databases, mark-up languages, referencing to memory and on disk, data storage in arrays and records and data transfer between modules within a programming language. Topics include the representation of data types in data dictionaries, the graphical representation of data using an Entity Relationship Diagram, how data is stored into separate entities using a relational database and the process of normalisation.

### **Systems development tools**

The course addresses hardware engineering practices and the skills required to create, modify and manage a computer hardware system, whether stand-alone or a network. It addresses the use of preventative maintenance and troubleshooting techniques to ensure the smooth running of computer hardware and communication systems. The course introduces methods used to create a communications network system, manage network servers and the connection and management of networking components, including network security procedures. It addresses the use of relevant system tools and a methodology to create or modify a database management information system and pieces of software using a high order computer language.

The course introduces basic methods for solving problems. This may involve the creation of an algorithm to represent the logic required to solve a problem; the use of design tools, to represent a solution to a problem before coding a computer program; the use of Entity Relationship Diagrams and data dictionaries in the creation of a database management system; the use of schematic diagrams in developing network and communication systems; and the use of a Gantt chart in the implementation of a computer system.

Topics include the different types of programming languages—first, second, third and fourth generation, procedural, non-procedural, object-oriented and scripting languages. Different categories of software, such as operating systems, network and application software and system tools are addressed. The basic constructs of sequence, selection and iteration when programming and how these are represented in a high level programming language are examined. Topics address the analysing and breaking up of a problem into small, self-contained units for which procedures or functions are then created in a programming language which includes the passing of parameters to procedures, functions and modules. It also includes the means by which records, files and databases in an application are accessed and an understanding of the operation of compilers and interpreters.

## **Course units**

Each unit is defined with a particular focus and a selection of learning contexts through which the specific unit content can be taught and learnt. The cognitive difficulty of the content increases with each stage and is referenced to the broad learning described in the outcome progressions. The pitch of the content for each stage is notional and there will be overlap between stages.

Stage 1 units provide bridging support and a practical and applied focus to help students develop skills required to be successful for Stage 2 units. The content is notionally pitched at levels 3 to 4.

Stage 2 units provide opportunities for applied learning but there is a focus more on academic learning. The content is notionally pitched at levels 4 to 6.

Stage 3 units provide opportunities to extend knowledge and understandings in challenging academic learning contexts. The content is notionally pitched at levels 6 to 8.

## Unit 1ACSC

The focus for this unit is the **personal use of computer systems**. It covers the knowledge and skills required to maintain a personal computer. Whilst the focus of this unit is on the hardware, the students are exposed to software for personal use, including the maintenance of an operating system, software for internet connection and software that allows students to write a sequence of simple instructions. Whilst considering personal needs, students examine the social, ethical and legal implications of personal computer use.

## Unit 1BCSC

The focus for this unit is the **personal use of communication and information systems**. It introduces a formal method for developing simple information systems, databases, networks and internet technologies. Students gain an understanding of the concepts and skills required to create and implement a system. They examine the social, ethical and legal implications of communication and information systems use.

## Unit 2ACSC

The focus for this unit is **developing systems solutions**. Students are introduced to the internal, interrelating components of computer-based systems in **an industry context**. They examine hardware and software design concepts and skills to meet a variety of computer-based challenges, using diagrammatic tools. Through the use of algorithms, students develop programming skills. Whilst considering industry requirements, they examine the social, ethical and legal implications of various solutions to industry problems.

## Unit 2BCSC

The focus for this unit is **developing database and communication systems solutions**. Students are introduced to networking concepts, **as applied to industry**. They examine a variety of systems, build on database and internet skills and gain an appreciation of how databases and internet and communication technologies are used in industry. Students create solutions involving databases and communications, exploring the ethical, legal and societal implications of these industry-based applications.

## Unit 3ACSC

The focus for this unit is the **design and development of software solutions**. Students develop a conceptual understanding of how a computer works and appreciate how large-scale systems are designed, developed and maintained. They gain the knowledge and skills to create software that will solve a range of problems. Students use algorithms and structured programming to design and implement software solutions. The students examine attitudes and values that lead to the creation and use of computer-based systems and their effect on society.

## Unit 3BCSC

The focus for this unit is the **design and development of database applications and communication systems**. Students consider communication systems, including security, protocols and the implications for web-based systems. They understand the design concepts and tools used to develop relational database systems. This takes students from the initial examination of data structures through to the creation of database applications using a current Database Management System (DBMS). Students consider the complex interactions between users, developers, the law, ethics and society when computer-based systems are used and developed.

## Time and completion requirements

The notional hours for each unit are 55 class contact hours. Units can be delivered typically in a semester or in a designated time period up to a year depending on the needs of the students. Pairs of units can also be delivered concurrently over a one year period. Schools are encouraged to be flexible in their timetabling in order to meet the needs of all of their students.

A unit is completed when all assessment requirements for that unit have been met. Only completed units will be recorded on a student's statement of results.

Refer to the WACE Manual for details about unit completion and course completion.

## Vocational Education Training information

Vocational Education Training (VET) is nationally recognised training that provides practical work skills and credit towards, or attainment of, a vocational education and training qualification.

When considering VET delivery in courses it is necessary to:

- refer to the WACE Manual, Section 6: Vocational Education Training, and
- contact education sector/systems representatives for information on operational issues concerning VET delivery options in schools.

### Australian Quality Training Framework (AQTF)

AQTF is the quality system that underpins the national vocational education and training sector and outlines the regulatory arrangements in states and territories. It provides the basis for a nationally consistent, high-quality VET system.

The AQTF Standards for Registered Training Organisations outline a set of auditable standards that must be met and maintained for registration as a training provider in Australia.

### **VET delivery**

VET can be delivered by schools providing they meet Australian Quality Training Framework (AQTF) requirements. Schools need to become a Registered Training Organisation (RTO) or work in partnership (auspicing arrangement) with an RTO to deliver training within the scope for which they are registered. If a school operates in partnership with an RTO, it will be the responsibility of the RTO to assure the quality of the training delivery and assessment. Qualifications identified in this course must be on the scope of registration of the RTO delivering or auspicing training.

Units of competency from selected training package qualifications have been taken into account during the development of this course.

Schools seeking to link delivery of this course with units of competency or qualification must read the information outlined in the relevant training package/s. This information can be found at the National Training Information Service website: [www.ntis.gov.au](http://www.ntis.gov.au).

### **National Training Package**

ICA05 Information and Communications Technology

#### **Qualifications**

ICA10105 Certificate I in Information Technology

ICA20105 Certificate II in Information Technology

Note: Any reference to qualifications and units of competency from training packages is correct at the time of accreditation.

## **Resources**

A detailed list of textbooks, teacher references, teacher guides and manuals can be found at [www.det.wa.edu.au/education/cmris/eval/curriculum/courses/](http://www.det.wa.edu.au/education/cmris/eval/curriculum/courses/)

# Assessment

Refer to the WACE Manual for policy and principles for both school-based assessment and examinations.

## School-based assessment

The three types of assessment in the table below are consistent with the teaching and learning strategies considered to be the most supportive of student achievement of the outcomes in the Computer Science course. The table provides details of the assessment types, including examples of different ways that they can be applied and the weighting range for each assessment type.

Teachers are to use the assessment table to develop their own assessment outlines.

An assessment outline needs to be developed for each class group enrolled in each unit of the course. This outline includes a range of assessment tasks that cover all assessment types and course outcomes with specific weightings. If units are delivered concurrently, assessment requirements must still be met for each unit.

In developing assessment outlines and teaching programs the following guidelines should be taken into account.

- All tasks should take into account teaching, learning and assessment principles from the Curriculum Framework.
- There is flexibility within the assessment framework for teachers to design school-based assessment tasks to meet the learning needs of students.
- Student responses may be communicated in any appropriate form e.g. written, oral, graphical, multimedia or various combinations of these.
- Student work submitted to demonstrate achievement of outcomes should only be accepted if the teacher can attest that, to the best of her/his knowledge, all uncited work is the student's own.
- Evidence collected for each unit should include tasks conducted under test conditions.

<b>Assessment table</b>			
<b>Weightings for types</b>			<b>Type of assessment</b>
<b>Stage 1</b>	<b>Stage 2</b>	<b>Stage 3</b>	
15–25%	15–25%	15–25%	<p><b>Investigation</b></p> <p>Research work in which students plan, conduct and communicate the findings of an investigation. The findings may be communicated in any appropriate form.</p> <p>Students are required to communicate ideas based on research, analysis and synthesis using appropriate conventions and forms. Substantiating evidence of research through use of scaffolding and note-taking, drafts, reference lists and so on.</p> <p>Types of evidence may include: written report, oral presentation (PowerPoint, video or audio) observation checklists, evaluation tools (self or peer) and/or journals.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 4.</i></p>
45–65%	35–50%	35–45%	<p><b>Production/practical</b></p> <p>Project work in which students explore ideas, develop and evaluate solutions and manage processes throughout production.</p> <p>Practical work in which students are required to complete tasks and/or exercises designed to develop and/or assess skills in using a programming language and a relational database management system. Programming skills include writing code, compiling, testing and debugging program code. Database skills include creating fields, data types and keys for tables and creating queries, forms and reports.</p> <p>Types of evidence may include product and/or practical observation checklists or evaluation tools (self or peer). Evidence may include a journal to show evidence of exploration and the development of ideas, reflection on learning processes and critical evaluation and modification of ideas.</p> <p>At least one practical programming task or one practical database task must be conducted under test conditions.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3.</i></p>
20–30%	35–50%	40–50%	<p><b>Response</b></p> <p>Students apply their knowledge and skills in analysing and responding to a series of stimuli or prompts.</p> <p>In-class and out of class activities demonstrating use and management of computer systems and an understanding of problem analysis, design, and the creation and evaluation of systems adhering to the required standards.</p> <p>Documentation of, and reflection on, own learning processes, enterprising capabilities and use of the technology process.</p> <p>Types of evidence may include: diagnostic, summative, formative test and examinations, practical exercises, responses to scenarios and problem-based exercises, observation checklists, journal and evaluation tools (self or peer).</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 2, 3 and 4.</i></p>

## Grades

Schools assign grades following the completion of the course unit. The following grades may be used:

Grade	Interpretation
A	Excellent achievement
B	High achievement
C	Satisfactory achievement
D	Limited achievement
E	Inadequate achievement

Preliminary Stage units are not graded. Achievement in these units is reported as either Completed or Not Completed.

Each grade is based on the student's overall performance for the course unit as judged by reference to a set of pre-determined standards. These standards are defined by grade descriptors.

Grade descriptors:

- describe the range of performances and achievement characteristics of grades A, B, C, D and E in a given stage of a course
- can be used at all stages of planning, assessment and implementation of courses, but are particularly important as a final point of reference in assigning grades
- are subject to continuing review by the Council.

The grade descriptors for this course can be accessed on the course page at [http://www.curriculum.wa.edu.au/internet/Senior\\_Secondary/Courses/Computer\\_Science](http://www.curriculum.wa.edu.au/internet/Senior_Secondary/Courses/Computer_Science)

## Examination details

There are separate examinations for Stage 2 pairs of units and Stage 3 pairs of units.

In their final year, students who are studying at least one Stage 2 pair of units (e.g. 2A/2B) or one Stage 3 pair of units (e.g. 3A/3B) will sit an examination in this course, unless they are exempt.

Each examination will assess the specific content, knowledge and skills described in the syllabus for the pair of units studied.

Details of the examinations in this course are prescribed in the examination design briefs (pages 23–25).

# UNIT 1ACSC

---

## Unit description

The focus for this unit is the **personal use of computer systems**. It gives students the knowledge and skills required to maintain a personal computer. Whilst the focus of this unit is on the hardware, the students are exposed to software for personal use, including the maintenance of an operating system, for internet connection and to write a sequence of simple instructions. Whilst considering personal needs, students examine the social, ethical and legal implications of personal computer use.

## Unit learning contexts

Within the broad area of the **personal use of computer systems**, teachers may choose one or more of the following contexts (this list is not exhaustive):

- home banking
- e-commerce
- online share trading
- home computer support.

## Unit content

This unit includes knowledge, understanding and skills to the degree of complexity described below:

Typically about 60 percent of the time would be spent on the tools area and the rest of the time divided between components and design, development and management areas.

## Components

### Hardware components

- identification of components to address processing (Central Processing Unit, input devices, output devices, primary memory, secondary memory/storage)
- identification of components to address communication functions (buses, expansion slots)
- storage types
  - magnetic (disk and tape)
  - optical (CD and DVD)
  - chip (USB flash drive)
- memory types (RAM, ROM, flash)
- peripheral devices (mouse, keyboard, scanner, printer, digital camera, microphone, speaker)
- ports (USB, firewire, PS2, Ethernet, serial)
- basic care and handling of hardware equipment including personal safety and proper use of components.

### Software components

- common application software (word processing, spreadsheet, database, graphics manipulation, desktop publishing, animation, CAD)
- internet software (browser, plug-in, email, FTP)
- operating systems and use of common peripheral devices
- basic maintenance and protection software (defragmentation, error checking, disk cleanup, backup, anti-malware).

## Design, development and management

### Systems

- hardware and software systems used in personal computing (applications, operating systems)
- interrelationship between users, hardware and software
- input, processing and output for personal computers
- installation of simple software and peripheral devices
- basic maintenance procedures to rectify simple difficulties
- introduction to planning, producing and evaluating a simple software system.

### Ethics, law and society

- ethical and legal issues in the personal use and storage of data
- measures an individual can take to help maintain data privacy and security e.g. personal firewall
- how user and developer needs and wants influence the choice, use and creation of personal computer systems
- ergonomics associated with computer use.

## Tools

### Data representation and organisation

- the concept of a data type
- program components e.g. inputs, processing, outputs *\*refer to Specifications booklet.*

### Systems development tools

- configuration of a home computer for peripherals and internet connection e.g. use of wizards
- simple programming language e.g. macros, visual programming language *\*refer to Specifications booklet.*

## VET units of competency

Units of competency may be delivered in appropriate learning contexts if all AQTF requirements are met. Some suggested units of competency suitable for integration are:

**ICAU1128B** Operate a personal computer

**ICAU2005B** Operate computer hardware

Note: Any reference to qualifications and units of competency from training packages is correct at the time of accreditation.

## Assessment

The three types of assessment in the table below are consistent with the teaching and learning strategies considered to be the most supportive of student achievement of the outcomes in the Computer Science course. The table provides details of the assessment type, examples of different ways that these assessment types can be applied and the weighting range for each assessment type.

Weighting Stage 1	Type of assessment
15–25%	<p><b>Investigation</b></p> <p>Research work in which students plan, conduct and communicate the findings of an investigation. The findings may be communicated in any appropriate form.</p> <p>Students are required to communicate ideas based on research, analysis and synthesis using appropriate conventions and forms. Substantiating evidence of research through use of scaffolding and note-taking, drafts, reference lists and so on.</p> <p>Types of evidence may include: written report, oral presentation (PowerPoint, video or audio) observation checklists, evaluation tools (self or peer) and/or journals.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 4.</i></p>
45–65%	<p><b>Production/practical</b></p> <p>Project work in which students explore ideas, develop and evaluate solutions and manage processes throughout production.</p> <p>Practical work in which students are required to complete tasks and/or exercises designed to develop and/or assess skills in using a programming language and a relational database management system. Programming skills include writing code, compiling, testing and debugging program code. Database skills include creating fields, data types and keys for tables and creating queries, forms and reports.</p> <p>Types of evidence may include product and/or practical observation checklists or evaluation tools (self or peer). Evidence may include a journal to show evidence of exploration and the development of ideas, reflection on learning processes and critical evaluation and modification of ideas.</p> <p>At least one practical programming task or one practical database task must be conducted under test conditions.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3.</i></p>
20–30%	<p><b>Response</b></p> <p>Students apply their knowledge and skills in analysing and responding to a series of stimuli or prompts.</p> <p>In-class and out of class activities demonstrating use and management of computer systems and an understanding of problem analysis, design, and the creation and evaluation of systems adhering to the required standards.</p> <p>Documentation of, and reflection on, own learning processes, enterprising capabilities and use of the technology process.</p> <p>Types of evidence may include: diagnostic, summative, formative test and examinations, practical exercises, responses to scenarios and problem-based exercises, observation checklists, journal and evaluation tools (self or peer).</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 2, 3 and 4.</i></p>

# UNIT 1BCSC

---

## Unit description

The focus for this unit is the **personal use of communication and information systems**. It introduces a formal method for developing simple information systems, databases, networks and internet technologies. Students gain an understanding of the concepts and skills required to create and implement a system. They examine the social, ethical and legal implications of communication and information systems use.

## Unit learning contexts

Within the broad area of the **use of personal communication and information systems**, teachers may choose one or more of the following contexts (this list is not exhaustive):

- small home office (SOHO) solutions
- personal communication
- home networking.

## Unit content

This unit includes knowledge, understanding and skills to the degree of complexity described below:

Typically about 60 percent of the time would be spent on the tools area and the rest of the time divided between components and design, development and management areas.

## Components

### Hardware components

- peer-to-peer networking
- identification of components to address communication process functions (modem, network interface card, switch, router, access point)
- key concepts, terminology and functions of common network components
  - data transmission rates (Mbps, Gbps)
  - wired data transmission media (twisted pair and optical fibre)
  - wireless transmission.

### Software components

- application software (browser, email, web authoring, scripting)
- internet connection and use of common peripheral devices
- maintenance procedures to rectify simple difficulties
- directory structures (folders, files)
- file types for web use (htm, html, pdf, jpg, gif, png, mp3, wav, swf, wmv).

## Design, development and management

### Systems

- techniques for developing simple personal databases and communication systems
- modification of an existing information system
- creation and administration of a simple network e.g. peer-to-peer, bluetooth
- configuration of a networked personal computer for peripherals (printer, scanner) and internet connection.

### Ethics, law and society

- reliability of internet data for personal use
- role of users and developers in maintaining the security of information transmitted through communication systems
- legal requirements and implications of information kept by various bodies about individuals.

## Tools

### Data representation and organisation

- data types used with database applications e.g. number (integer, real, autonumber), text/string, Boolean (yes/no), date, currency
- flow of data in an information system
- components of a single table database (field, record, file)
- awareness of internet protocols (HTTP, TCP/IP, SMTP, POP3 and FTP).

### Systems development tools

- single table database application to create tables, filters, sorts, simple queries, forms and reports *\*refer to Specifications booklet*
- comparison of web-based construction tools
- use of simple web tools to create simple linked web pages
- use of content management system tools to upload web pages to a web server.

## VET units of competency

Units of competency may be delivered in appropriate learning contexts if all AQTF requirements are met. Some suggested units of competency suitable for integration are:

**ICAU1131B** Operate a database application

**ICAU1133B** Send and retrieve information using web browsers and email

Note: Any reference to qualifications and units of competency from training packages is correct at the time of accreditation.

# Assessment

The three types of assessment in the table below are consistent with the teaching and learning strategies considered to be the most supportive of student achievement of the outcomes in the Computer Science course. The table provides details of the assessment type, examples of different ways that these assessment types can be applied and the weighting range for each assessment type.

Weighting Stage 1	Type of assessment
15–25%	<p><b>Investigation</b></p> <p>Research work in which students plan, conduct and communicate the findings of an investigation. The findings may be communicated in any appropriate form.</p> <p>Students are required to communicate ideas based on research, analysis and synthesis using appropriate conventions and forms. Substantiating evidence of research through use of scaffolding and note-taking, drafts, reference lists and so on.</p> <p>Types of evidence may include: written report, oral presentation (PowerPoint, video or audio) observation checklists, evaluation tools (self or peer) and/or journals.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 4.</i></p>
45–65%	<p><b>Production/practical</b></p> <p>Project work in which students explore ideas, develop and evaluate solutions and manage processes throughout production.</p> <p>Practical work in which students are required to complete tasks and/or exercises designed to develop and/or assess skills in using a programming language and a relational database management system. Programming skills include writing code, compiling, testing and debugging program code. Database skills include creating fields, data types and keys for tables and creating queries, forms and reports.</p> <p>Types of evidence may include product and/or practical observation checklists or evaluation tools (self or peer). Evidence may include a journal to show evidence of exploration and the development of ideas, reflection on learning processes and critical evaluation and modification of ideas.</p> <p>At least one practical programming task or one practical database task must be conducted under test conditions.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3.</i></p>
20–30%	<p><b>Response</b></p> <p>Students apply their knowledge and skills in analysing and responding to a series of stimuli or prompts.</p> <p>In-class and out of class activities demonstrating use and management of computer systems and an understanding of problem analysis, design, and the creation and evaluation of systems adhering to the required standards.</p> <p>Documentation of, and reflection on, own learning processes, enterprising capabilities and use of the technology process.</p> <p>Types of evidence may include: diagnostic, summative, formative test and examinations, practical exercises, responses to scenarios and problem-based exercises, observation checklists, journal and evaluation tools (self or peer).</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 2, 3 and 4.</i></p>

# UNIT 2ACSC

---

## Unit description

The focus for this unit is **developing systems solutions**. Students are introduced to the internal, interrelating components of computer-based systems in **an industry context**. They examine hardware and software design concepts and skills to meet a variety of computer-based challenges, using diagrammatic tools. Through the use of algorithms, students develop programming skills. Whilst considering industry requirements, they examine the social, ethical and legal implications of various solutions to industry problems.

## Unit learning contexts

Within the broad area of **developing system solutions**, teachers may choose one or more of the following contexts (this list is not exhaustive):

- fashion industry
- landscape design
- manufacturing
- robotics
- virtual reality
- entertainment
- sports science.

## Unit content

This unit includes knowledge, understanding and skills to the degree of complexity described below:

Typically about 60 percent of the time would be spent on the tools area and the rest of the time divided between components and design, development and management areas.

## Components

### Hardware components

- identification of components in industry systems to address processing, input, output, memory/storage and communication functions:
  - central processing unit description and purpose (registers, arithmetic logic unit, control unit, program counter, clock)
  - input devices (mouse, keyboard, tablet, touch screen, bar code scanner, RFID reader, microphone, video capture)
  - output devices (monitor, printer, plotter, speaker)
  - primary storage (RAM, ROM, cache)
  - secondary storage types (magnetic, optical, flash, DAT)
  - buses (data, address, control).

- standards in hardware components
  - storage capacities (bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte)
  - processor and bus speeds (megahertz, gigahertz)
  - graphics displays (resolution)
  - components from different manufacturers working together.
- evolution of business hardware
- comparing performance using benchmarking.

### Software components

- business application software (accounting, database, presentation, project management, spreadsheet, word processing)
- graphics and multimedia software (CAD, desktop publishing, image editing, multimedia authoring, video and audio editing, web page authoring)
- generations of programming languages (machine, assembler, high level)
- operating systems (Linux, Unix, Windows, MacOS)
- utility software (disk defragmenter, anti-malware, compression, encryption, virus checking software)
- backup and restore.

## Design, development and management

### Systems

- different types of business systems (transaction processing, groupware, expert systems, decision-support systems)
- benefits and economic considerations for implementation of industry type systems
- methods for developing systems including the Systems Development Life Cycle and prototyping *\*refer to Specifications booklet*.

### Ethics, law and society

- user and developer concerns about inclusivity and empowerment
- monitoring employee use of computer-based systems (email and internet use)
- licensing requirements for software e.g. single-use, multi-use, network and restricted and unrestricted site
- cost benefit of open source, shareware, commercial software.

## Tools

### Data representation and organisation *\*refer to Specifications booklet*

- program components and constructs (variables, constants, data types, statements, sequence, selection and repetition)
- number systems (decimal, binary, hexadecimal)
- encoding (ASCII, Unicode).

**Systems development tools** \*refer to Specifications booklet

- use of simple algorithms and a programming language to develop computer-based systems for industry
- trace tables for desk checking, testing and debugging
- Data Flow Diagrams (context diagram, Level 0 with 3 to 5 processes).

## VET units of competency

Units of competency may be delivered in appropriate learning contexts if all AQTF requirements are met. Some suggested units of competency suitable for integration are:

**ICAU2005B Operate computer hardware**

Note: Any reference to qualifications and units of competency from training packages is correct at the time of accreditation.

## Assessment

The three types of assessment in the table below are consistent with the teaching and learning strategies considered to be the most supportive of student achievement of the outcomes in the Computer Science course. The table provides details of the assessment type, examples of different ways that these assessment types can be applied and the weighting range for each assessment type.

Weighting Stage 2	Type of assessment
15–25%	<p><b>Investigation</b></p> <p>Research work in which students plan, conduct and communicate the findings of an investigation. The findings may be communicated in any appropriate form.</p> <p>Students are required to communicate ideas based on research, analysis and synthesis using appropriate conventions and forms. Substantiating evidence of research through use of scaffolding and note-taking, drafts, reference lists and so on.</p> <p>Types of evidence may include: written report, oral presentation (PowerPoint, video or audio) observation checklists, evaluation tools (self or peer) and/or journals.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 4.</i></p>
35–50%	<p><b>Production/practical</b></p> <p>Project work in which students explore ideas, develop and evaluate solutions and manage processes throughout production.</p> <p>Practical work in which students are required to complete tasks and/or exercises designed to develop and/or assess skills in using a programming language and a relational database management system. Programming skills include writing code, compiling, testing and debugging program code. Database skills include creating fields, data types and keys for tables and creating queries, forms and reports.</p> <p>Types of evidence may include product and/or practical observation checklists or evaluation tools (self or peer). Evidence may include a journal to show evidence of exploration and the development of ideas, reflection on learning processes and critical evaluation and modification of ideas.</p> <p>At least one practical programming task or one practical database task must be conducted under test conditions.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3.</i></p>
35–50%	<p><b>Response</b></p> <p>Students apply their knowledge and skills in analysing and responding to a series of stimuli or prompts.</p> <p>In-class and out of class activities demonstrating use and management of computer systems and an understanding of problem analysis, design, and the creation and evaluation of systems adhering to the required standards.</p> <p>Documentation of, and reflection on, own learning processes, enterprising capabilities and use of the technology process.</p> <p>Types of evidence may include: diagnostic, summative, formative test and examinations, practical exercises, responses to scenarios and problem-based exercises, observation checklists, journal and evaluation tools (self or peer).</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 2, 3 and 4.</i></p>

# UNIT 2BCSC

---

## Unit description

The focus for this unit is **developing database and communication systems solutions**. Students are introduced to networking concepts, **as applied to industry**. They examine a variety of systems, build on database and internet skills and gain an appreciation of how databases and internet and communication technologies are used in industry. Students create solutions involving databases and communications, exploring the ethical, legal and societal implications of these industry-based applications.

## Unit learning contexts

Within the broad area of **developing database and communication systems solutions**, teachers may choose one or more of the following contexts (this list is not exhaustive):

- biometrics
- farming and agriculture
- travel and tourism
- architectural design
- pharmaceutical enterprise.

## Unit content

This unit includes knowledge, understanding and skills to the degree of complexity described below:

Typically about 60 percent of the time would be spent on the tools area and the rest of the time divided between components and design, development and management areas.

## Components

### Hardware components

- identification of components to address communication process functions (sending devices, communications devices, channels or pathways, receiving devices, messages, protocols)
- key concepts, terminology, functions, capabilities and limitations of common components in communication systems:
  - types of networks (PAN, LAN, WAN)
  - factors affecting the design of networks
  - digital and analogue transmissions
  - network devices (network interface card, switch, router, modem, access point)
  - protocols (TCP/IP, WAP, Ethernet, FTP, HTTP, SMTP)
  - transmission rates (bps, Kbps, Mbps, Gbps) and the need for bandwidth

- transmission media:
  - copper (coaxial, twisted pair)
  - optical fibre (single-mode, multi-mode)
  - wireless (infrared, cellular, radio, microwave, satellite).
- purpose of subnets
- network standards (Ethernet standards 802.3 and 802.11, Bluetooth, IrDA, RFID).
- network topologies for LANs (star, ring, bus)
- client/server and peer-to-peer
- security of information over the internet (firewalls, authentication and encryption).

### Software components

- database systems (flat file, relational)
- features and functions of web interfaces to databases
- features of network operating systems (administration, resource management, security).

## Design, development and management

### Systems

- database system components (tables, queries, forms, reports)
- web-based business systems (intranet, extranet, internet, content management system)
- database system documentation (data dictionary)
- TCP/IP model \* *refer to Specifications booklet*.

### Ethics, law and society

- legal requirements for the collection, storage and use of information
- rights and responsibilities of users, developers and businesses
- data security and privacy
- demographics of computer use in Australia.

## Tools

### Data representation and organisation

- Database Management Systems to manipulate multiple tables
- normalisation concepts (normalisation, relation, primary and foreign key, efficiency of stored data, update anomaly, data integrity, redundancy)
- use of 1:M relationship to represent linked tables
- comparison of sequential and indexed file processing.

### Systems development tools

- use of diagrams to represent network devices and network design
- introduction to Entity Relationship Diagrams \**refer to Specifications booklet*
- relational database application to create 2 to 3 linked tables, queries, forms and reports.

## VET units of competency

Units of competency may be delivered in appropriate learning contexts if all AQTF requirements are met. Some suggested units of competency suitable for integration are:

- ICAW2001B** Work effectively in an IT environment
- ICAW2002B** Communicate in the workplace
- ICAU2013B** Integrate commercial computing packages
- ICPMM263B** Access the internet

Note: Any reference to qualifications and units of competency from training packages is correct at the time of accreditation.

## Assessment

The three types of assessment in the table below are consistent with the teaching and learning strategies considered to be the most supportive of student achievement of the outcomes in the Computer Science course. The table provides details of the assessment type, examples of different ways that these assessment types can be applied and the weighting range for each assessment type.

Weighting Stage 2	Type of assessment
15–25%	<p><b>Investigation</b></p> <p>Research work in which students plan, conduct and communicate the findings of an investigation. The findings may be communicated in any appropriate form.</p> <p>Students are required to communicate ideas based on research, analysis and synthesis using appropriate conventions and forms. Substantiating evidence of research through use of scaffolding and note-taking, drafts, reference lists and so on.</p> <p>Types of evidence may include: written report, oral presentation (PowerPoint, video or audio) observation checklists, evaluation tools (self or peer) and/or journals.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 4.</i></p>
35–50%	<p><b>Production/practical</b></p> <p>Project work in which students explore ideas, develop and evaluate solutions and manage processes throughout production.</p> <p>Practical work in which students are required to complete tasks and/or exercises designed to develop and/or assess skills in using a programming language and a relational database management system. Programming skills include writing code, compiling, testing and debugging program code. Database skills include creating fields, data types and keys for tables and creating queries, forms and reports.</p> <p>Types of evidence may include product and/or practical observation checklists or evaluation tools (self or peer). Evidence may include a journal to show evidence of exploration and the development of ideas, reflection on learning processes and critical evaluation and modification of ideas.</p> <p>At least one practical programming task or one practical database task must be conducted under test conditions.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3. Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3.</i></p>
35–50%	<p><b>Response</b></p> <p>Students apply their knowledge and skills in analysing and responding to a series of stimuli or prompts.</p> <p>In-class and out of class activities demonstrating use and management of computer systems and an understanding of problem analysis, design, and the creation and evaluation of systems adhering to the required standards.</p> <p>Documentation of, and reflection on, own learning processes, enterprising capabilities and use of the technology process.</p> <p>Types of evidence may include: diagnostic, summative, formative test and examinations, practical exercises, responses to scenarios and problem-based exercises, observation checklists, journal and evaluation tools (self or peer).</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 2, 3 and 4.</i></p>

# UNIT 3ACSC

---

## Unit description

The focus for this unit is the **design and development of software solutions**. Students develop a conceptual understanding of how a computer works and appreciate how large-scale systems are designed, developed and maintained. They gain the knowledge and skills to create software that solves a range of problems. Students use algorithms and structured programming to design and implement software solutions. The students examine attitudes and values that lead to the creation and use of computer-based systems and their effect on society.

## Unit learning contexts

Within the broad area of the **design and development of software solutions**, teachers may choose one or more of the following contexts (this list is not exhaustive):

- financial institutions
- computer game development
- autonomous robotics
- security systems
- forensic science
- expert systems.

## Unit content

This unit includes knowledge, understanding and skills to the degree of complexity described below:

Typically about 60 percent of the time would be spent on the tools area and the rest of the time divided between components and design, development and management areas.

## Components

### Hardware components

- computer architecture
  - role of the central processing unit (arithmetic logic unit, control unit, registers, program counter, cache), system clock and data bus in the fetch-execute cycle
  - multi-core processor architectures e.g. dual-core, quad-core
  - multi-processor and parallel system architecture.
- fault diagnosis e.g. selective component removal/replacement
- hardware preventative maintenance e.g. inspection, cleaning and repair
- backup recovery systems (RAID, backup, snapshot imaging)
- virtualisation.

### Software components

- purpose of operating systems
- distinguish between batch systems and real-time systems
- identify functions of operating systems (scheduling, managing concurrency, managing memory, managing devices, file systems)
- different programming languages (procedural, non-procedural, object-oriented, event-driven, platform-independent, visual, scripting).

## Design, development and management

### Systems

- project management tasks (scheduling, budgeting, tracking)
- project management tools (Gantt charts, milestone charts, PERT/CPM charts)
- use of Systems Development Life Cycle *\*refer to Specifications booklet*
- variations of SDLC (top-down, bottom-up, prototyping and rapid application development).

### Ethics, law and society

- interface evaluation and inclusivity
- professional ethics and personal skills of developers and users
- present and future impacts of computer-based systems on society.

### Tools *\* refer to Specifications booklet*

### Data representation and organisation

- programming constructs (constants, variables, statements, control structures, modularisation)
- fundamentals of data structures (one-dimensional arrays, records and files)
- data storage and access (content management systems and portals).

### Systems development tools

- structured programming using modularisation and parameter passing including stubs, subs, functions, scope of identifiers, structure charts
- algorithmic and programming techniques including documentation, testing and debugging logic, syntax and run-time errors
- levelled Data Flow Diagrams (context diagram, Level 0, Level 1 DFD).

# Assessment

The three types of assessment in the table below are consistent with the teaching and learning strategies considered to be the most supportive of student achievement of the outcomes in the Computer Science course. The table provides details of the assessment type, examples of different ways that these assessment types can be applied and the weighting range for each assessment type.

Weighting Stage 3	Type of assessment
15–25%	<p><b>Investigation</b></p> <p>Research work in which students plan, conduct and communicate the findings of an investigation. The findings may be communicated in any appropriate form.</p> <p>Students are required to communicate ideas based on research, analysis and synthesis using appropriate conventions and forms. Substantiating evidence of research through use of scaffolding and note-taking, drafts, reference lists and so on.</p> <p>Types of evidence may include: written report, oral presentation (PowerPoint, video or audio) observation checklists, evaluation tools (self or peer) and/or journals.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 4.</i></p>
35–45%	<p><b>Production/practical</b></p> <p>Project work in which students explore ideas, develop and evaluate solutions and manage processes throughout production.</p> <p>Practical work in which students are required to complete tasks and/or exercises designed to develop and/or assess skills in using a programming language and a relational database management system. Programming skills include writing code, compiling, testing and debugging program code. Database skills include creating fields, data types and keys for tables and creating queries, forms and reports.</p> <p>Types of evidence may include product and/or practical observation checklists or evaluation tools (self or peer). Evidence may include a journal to show evidence of exploration and the development of ideas, reflection on learning processes and critical evaluation and modification of ideas.</p> <p>At least one practical programming task or one practical database task must be conducted under test conditions.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3.</i></p>
40–50%	<p><b>Response</b></p> <p>Students apply their knowledge and skills in analysing and responding to a series of stimuli or prompts.</p> <p>In-class and out of class activities demonstrating use and management of computer systems and an understanding of problem analysis, design, and the creation and evaluation of systems adhering to the required standards.</p> <p>Documentation of, and reflection on, own learning processes, enterprising capabilities and use of the technology process.</p> <p>Types of evidence may include: diagnostic, summative, formative test and examinations, practical exercises, responses to scenarios and problem-based exercises, observation checklists, journal and evaluation tools (self or peer).</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 2, 3 and 4.</i></p>

# UNIT 3BCSC

---

## Unit description

The focus for this unit is the **design and development of database applications and communication systems**. Students consider communication systems, including security, protocols and the implications for web-based systems. They understand the design concepts and tools used to develop relational database systems. This takes students from the initial examination of data structures through to the creation of database applications using a current Database Management System (DBMS). Students consider the complex interactions between users, developers, the law, ethics and society when computer-based systems are used and developed.

## Unit learning contexts

Within the broad area of the **design and development of database applications and communication systems**, teachers may choose one or more of the following contexts (this list is not exhaustive):

- aerospace
- defence
- government
- e-shopping systems
- booking systems
- decision support systems
- communication systems.

## Unit content

This unit includes knowledge, understanding and skills to the degree of complexity described below:

Typically about 60 percent of the time would be spent on the tools area and the rest of the time divided between components and design, development and management areas.

## Components

### Hardware components

- computer architecture required for communications
  - network devices (router, switch, firewall, modem, network interface card, wireless access point)
  - network media (twisted pair, coaxial, fibre optic, wireless)
  - network control protocols (CSMA/CD, CSMA/CA)
  - digital and analogue data transmission
  - error detection/correction in digital data transmission.

- network topologies and protocols for LANs
  - network types (PAN, HAN, LAN, MAN, WAN, GAN)
  - intranet, extranet and the internet
  - network topologies (star, bus, ring, point-to-point, hybrid)
  - communications protocols and standards
    - wireless (Ethernet 802.11x, WAP, Bluetooth)
    - wired (Ethernet 802.3, token ring 802.5)
    - TCP/IP and DNS.
- client server technology
- LAN and WAN structures (subnets)
- storage area networks (fibre channel, iSCSI).

### Software components

- types of databases (distributed, centralised)
- data warehouses, data marts and data mining
- protocols and security issues in network devices
- encryption methods e.g. private key encryption, public key encryption, PGP
- connection of databases to the internet e.g. querying a web database through use of ODBC, using SQL, PHP, VB.net.

## Design, development and management

### Systems

- function and use of database management systems (data definition, data integrity, data manipulation, data security)
- network interconnectivity (OSI model, TCP/IP model) *\*refer to Specifications booklet*
- database interconnectivity (ODBC)
- convergence of technologies e.g. VoIP and using the Internet for voice telephony, web casting of radio and TV programs
- development and management of database systems using data normalisation, data dictionary definitions, entity relationship diagrams, data flow diagrams
  - multiple table structure to store data
  - visual interface to assist user access
  - queries, forms, reports to display and extract data for users.

### Ethics, law and society

- security implications, encryption and secure websites
- user and developer codes of conduct when using/handling data in databases and networks
- privacy and legal implications of personal data
- influence of changing technologies on employment, entertainment and information
- global reliance on computer systems and the connectivity the internet provides.

## Tools \* refer to Specifications booklet

### Data representation and organisation

- normalisation to third normal form
- resolving many to many M:N relationships to develop a multi-table relational database system
- physical storage of data (online, distributed, local databases).

### Systems development tools

- LAN and WAN network diagrams
- database modelling tools including Entity Relationship Diagrams and normalisation
- use of relational database application to create multiple linked tables, queries, forms and reports.

## Assessment

The three types of assessment in the table below are consistent with the teaching and learning strategies considered to be the most supportive of student achievement of the outcomes in the Computer Science course. The table provides details of the assessment type, examples of different ways that these assessment types can be applied and the weighting range for each assessment type.

Weighting Stage 3	Type of assessment
15–25%	<p><b>Investigation</b></p> <p>Research work in which students plan, conduct and communicate the findings of an investigation. The findings may be communicated in any appropriate form.</p> <p>Students are required to communicate ideas based on research, analysis and synthesis using appropriate conventions and forms. Substantiating evidence of research through use of scaffolding and note-taking, drafts, reference lists and so on.</p> <p>Types of evidence may include: written report, oral presentation (PowerPoint, video or audio) observation checklists, evaluation tools (self or peer) and/or journals.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 4.</i></p>
35–45%	<p><b>Production/practical</b></p> <p>Project work in which students explore ideas, develop and evaluate solutions and manage processes throughout production.</p> <p>Practical work in which students are required to complete tasks and/or exercises designed to develop and/or assess skills in using a programming language and a relational database management system. Programming skills include writing code, compiling, testing and debugging program code. Database skills include creating fields, data types and keys for tables and creating queries, forms and reports.</p> <p>Types of evidence may include product and/or practical observation checklists or evaluation tools (self or peer). Evidence may include a journal to show evidence of exploration and the development of ideas, reflection on learning processes and critical evaluation and modification of ideas.</p> <p>At least one practical programming task or one practical database task must be conducted under test conditions.</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 1, 2 and 3.</i></p>
40–50%	<p><b>Response</b></p> <p>Students apply their knowledge and skills in analysing and responding to a series of stimuli or prompts.</p> <p>In-class and out of class activities demonstrating use and management of computer systems and an understanding of problem analysis, design, and the creation and evaluation of systems adhering to the required standards.</p> <p>Documentation of, and reflection on, own learning processes, enterprising capabilities and use of the technology process.</p> <p>Types of evidence may include: diagnostic, summative, formative test and examinations, practical exercises, responses to scenarios and problem-based exercises, observation checklists, journal and evaluation tools (self or peer).</p> <p><i>Best suited to the collection of evidence of student achievement of Outcomes 2, 3 and 4.</i></p>

For teaching from 2010

**Examination details  
Stage 2 and Stage 3**

For teaching from 2010

# Computer Science Examination design brief Stage 2

## Time allowed

Reading time before commencing work: ten minutes  
Working time for paper: three hours

## Permissible items

Standard items: pens, pencils, eraser, correction fluid, ruler, highlighters  
Special items: non-programmable calculators, MATHOMAT and/or Mathaid and/or any system flowchart template

Section	Supporting information
<p><b>Section One</b> <b>Multiple-choice</b> 10–20% of the examination 20 questions Suggested working time: 25 minutes</p>	<p>Questions in this section could require the candidate to interpret stimulus material including diagrams, tables, algorithm segments and screen captures of databases or programs.</p> <p>The candidate is required to indicate the answers on a separate multiple-choice answer sheet by selecting an appropriate response from a choice of four alternatives.</p>
<p><b>Section Two</b> <b>Short answer</b> 30–40% of the examination 15–20 questions Suggested working time: 65 minutes</p>	<p>Questions require the candidate to demonstrate an understanding of how the design and development of computer-based systems influences and impacts on individuals and communities.</p> <p>The candidate could be required to explain concepts, apply knowledge, analyse and interpret data, respond to stimulus materials or design solutions to scenarios.</p> <p>Questions could relate to stimulus material or a scenario and are structured as a range of closed and open items that increase in complexity. Stimulus material could include diagrams, short excerpts from newspaper or journal articles, algorithm segments and screen captures of databases or programs.</p>
<p><b>Section Three</b> <b>Extended answer</b> 40–55% of the examination 4–6 questions Suggested working time: 90 minutes</p>	<p>The candidate could be required to apply critical thinking skills in the analysis and interpretation of stimulus materials, which could include common scenarios.</p> <p>Questions could include scaffolded questions with parts that relate to scenarios, extended algorithms, database designs, diagrams and tables.</p> <p>The candidate could be required to complete or devise clearly labelled or annotated diagrams including context and Level 0 and Level 1 data flow diagrams.</p>

# Computer Science Examination design brief Stage 3

## Time allowed

Reading time before commencing work: ten minutes  
Working time for paper: three hours

## Permissible items

Standard items: pens, pencils, eraser, correction fluid, ruler, highlighters  
Special items: non-programmable calculators, MATHOMAT and/or Mathaid and/or any system flowchart template

Section	Supporting information
<p><b>Section One</b> <b>Short answer</b> 40% of the written exam 25–30 questions Suggested working time: 70 minutes</p>	<p>The candidate is assessed on their knowledge and understanding of computer-based systems and how these are continually evolving to meet changing industry and societal needs.</p> <p>The candidate could be required to explain concepts, apply knowledge, analyse and interpret data, respond to stimulus materials or design solutions to scenarios.</p> <p>Questions could be a range of closed and open items and could relate to stimulus materials or scenarios. Stimulus materials could include diagrams, short excerpts from newspaper or journal articles, algorithm segments and screen captures of databases or programs.</p>
<p><b>Section Two</b> <b>Extended answer</b> 60% of the written exam 4–6 questions Suggested working time: 110 minutes</p>	<p>Questions require the candidate to demonstrate a depth of knowledge of computer-based systems, as well as the constraints and possibilities resulting from vibrant environments.</p> <p>The candidate could be required to apply critical thinking skills, analyse and interpret complex data, respond to scenarios, devise clearly labelled diagrams, design sophisticated solutions or parts of solutions.</p> <p>Questions could be scaffolded or sectionalised and could relate to common scenarios, extended algorithms, multiple table database designs, diagrams, tables. Diagrams could include those related to project management tools, computer systems, networking, context, Level 0 and Level 1 data flow diagrams.</p>

For teaching from 2010

## **Appendix 1: Outcome Progressions**

For teaching from 2010

## Outcome progressions

<b>Outcome 1: Technology process</b> Students apply a technology process to develop computer-based systems.						
	<b>Level 3</b>	<b>Level 4</b>	<b>Level 5</b>	<b>Level 6</b>	<b>Level 7</b>	<b>Level 8</b>
	Students apply a sequence of steps to generate a plan, perform specified production processes and evaluate the final solution.	Students generate a design, organise and implement production processes, within given constraints, assessing how well the design and solution meet the design requirements.	Students include options in the design proposals, organise and implement production processes to own specifications and use given testing criteria to assess the effectiveness of the solution and production processes.	Students justify options in the design proposals, organise, implement and adjust detailed production processes and use a range of testing criteria to monitor and evaluate the effectiveness of the designs, solutions and processes.	Students devise detailed design and production proposals that show how ideas and strategies are developed; collaborate, monitor and adjust the production process and implement an ongoing evaluation process, presenting conclusions.	Students analyse implications and benefits to devise detailed proposals that justify the merits of different options; monitor, adjust, negotiate and document production processes; apply qualitative and quantitative testing in the ongoing evaluation and monitoring process; and consider current and future impacts.
<b>Students:</b>						
<ul style="list-style-type: none"> <li>investigate ideas and generate proposals.</li> <li>develop solutions that meet specifications and recognised standards.</li> <li>evaluate computer-based solutions.</li> </ul>	<ul style="list-style-type: none"> <li>investigate ideas, based on identified system requirements and develop designs using drawings and text.</li> <li>produce a solution following given standards and steps.</li> <li>make a comparison between the solution and design plan, using given criteria.</li> </ul>	<ul style="list-style-type: none"> <li>investigate and design a proposal for a computer-based system solution and communicate these ideas using graphical representations and descriptions using common computer-related terms.</li> <li>organise and implement production processes within given constraints.</li> <li>evaluate how the design plan and solution meet design requirements, using given testing criteria.</li> </ul>	<ul style="list-style-type: none"> <li>investigate and devise design proposals for computer-based system solutions, considering options, and communicate these ideas using graphical representations, simple algorithms and suitable computer-related terms.</li> <li>organise and implement own production schedules and processes according to the production proposal.</li> <li>determine the effectiveness of the designs, production process and solution, using given testing criteria in an evaluation process.</li> </ul>	<ul style="list-style-type: none"> <li>investigate and devise design proposals for computer-based system solutions that take account of a number of options, justify choices made and communicate these proposals using systems design tools, algorithms and technical language.</li> <li>organise, implement and adjust production schedules and processes according to detailed production plans to produce a functional solution that meets requirements.</li> <li>use a range of testing criteria and include these in an evaluation process to monitor and evaluate the effectiveness of the designs, processes and solution.</li> </ul>	<ul style="list-style-type: none"> <li>investigate and devise design and production proposals that show a development of ideas and strategies and communicate these proposals using system design and project management tools, complex algorithms and technical language.</li> <li>demonstrate efficient use of time and resources, consulting with others when organising, implementing and adjusting production processes to produce documented, functional systems.</li> <li>implement an ongoing evaluation process, using recognised techniques to monitor, evaluate and adjust the designs, processes and product, presenting evaluations and conclusions in a written report.</li> </ul>	<ul style="list-style-type: none"> <li>analyse implications and benefits to devise detailed design and production proposals for complex systems, justifying the merits of different options and communicating these proposals using complex system design and project management tools, algorithms and high level technical language.</li> <li>demonstrate optimum use of time, facilities and resources and negotiate with others when organising, implementing and adjusting production processes to produce fully documented systems.</li> <li>apply qualitative and quantitative testing techniques, based on recognised standards in an ongoing evaluation and monitoring process and present conclusions in a report that considers current and future impacts.</li> </ul>

## Outcome progressions

<b>Outcome 2: Knowledge and understanding of computer-based systems</b> Students understand the design, application and interactions of hardware and software in computer-based systems.						
	<b>Level 3</b>	<b>Level 4</b>	<b>Level 5</b>	<b>Level 6</b>	<b>Level 7</b>	<b>Level 8</b>
	Students understand that computer-based systems are composed of elements that interact for a particular purpose.	Students understand that the elements in computer-based systems interact and can be controlled.	Students understand the diversity of elements, their internal and external interactions and how they can be controlled and maintained.	Students understand the function of computer-based system components, their interactions to control and maintain systems and that these technologies are evolving.	Students understand the principles, structures, logic, organisation and control of components in computer-based systems and how these technologies are continually evolving to meet changing needs.	Students understand the theories underpinning the function, interaction and maintenance of complex computer-based systems and components; and the constraints and possibilities resulting from the dynamic environment in which these complex technologies evolve.
<b>Students:</b>						
<ul style="list-style-type: none"> <li>understand the appropriate selection and application of computer-based system components.</li> <li>understand the nature of the interactions between the elements of computer-based systems.</li> <li>understand the concepts associated with computer-based systems.</li> </ul>	<ul style="list-style-type: none"> <li>understand that hardware components and software are used for specific purposes.</li> <li>understand that hardware and software elements work together for a particular purpose.</li> <li>understand specified concepts and their application.</li> </ul>	<ul style="list-style-type: none"> <li>understand the features, selection and use of hardware and software components.</li> <li>understand that hardware and software elements interact to create a simple computer-based system.</li> <li>understand concepts describing how components interact and are controlled.</li> </ul>	<ul style="list-style-type: none"> <li>understand the diversity of hardware and software components, their features, selection and appropriate use.</li> <li>understand how hardware and software elements interact to create a computer-based system that interfaces with the external environment.</li> <li>understand concepts to explain how components and systems operate and are maintained.</li> </ul>	<ul style="list-style-type: none"> <li>understand how hardware and software components function to enable the appropriate selection and use of these components.</li> <li>understand the complexity of interactions between elements in computer-based systems and their effect on the operation and use of these systems.</li> <li>understand principles that explain how components are controlled, maintained and evolve.</li> </ul>	<ul style="list-style-type: none"> <li>understand the complex structures and function of hardware and software components, justifying their selection and use.</li> <li>understand underlying principles of the interacting elements of a computer-based system and how these are used to control system performance.</li> <li>understand structures, principles and logic that underpin the operation, control and evolution of computer-based systems.</li> </ul>	<ul style="list-style-type: none"> <li>understand the theories underpinning the design and operation of hardware and software components and how to apply these in an optimal system to meet the changing requirements of systems.</li> <li>understand methodologies to critically evaluate the interaction of complex computer-based system elements in order to optimise system performance.</li> <li>understand theories that underpin the development, control and evolution of complex computer-based information and communication systems to critically evaluate the interaction.</li> </ul>

## Outcome progressions

<b>Outcome 3: Skills for computer-based systems</b> Students apply skills to maintain, adapt or develop computer-based systems.						
	<b>Level 3</b>	<b>Level 4</b>	<b>Level 5</b>	<b>Level 6</b>	<b>Level 7</b>	<b>Level 8</b>
	Students use skills to apply provided conventions and follow specified procedures.	Students work within given constraints applying skills to solve simple problems, using a limited set of conventions, resources and procedures.	Students apply skills to solve predictable challenges, selecting and using conventions, resources and procedures.	Students apply skills that partition a problem, consider alternatives that incorporate appropriate industry conventions, standards and procedures.	Students apply high-level skills that justify alternatives, and demonstrate flexibility when refining and documenting solutions.	Students apply complex skills that take account of the diversity of standards and the complexity and dynamic nature of computer-based systems.
<b>Students:</b>						
<ul style="list-style-type: none"> <li>• <b>apply a range of problem-solving techniques when maintaining or developing computer-based systems.</b></li> <li>• <b>apply a range of conventions and standards when implementing a maintenance or development solution.</b></li> <li>• <b>apply organisational skills to identify and use appropriate hardware and software resources when maintaining or developing a computer-based system.</b></li> </ul>	<ul style="list-style-type: none"> <li>• apply a specified sequence of steps to devise a solution.</li> <li>• apply simple, provided conventions.</li> <li>• use specified resources following given procedures.</li> </ul>	<ul style="list-style-type: none"> <li>• apply simple problem-solving strategies to familiar problems.</li> <li>• apply a limited set of conventions working within given constraints.</li> <li>• identify resources from a limited set and use familiar procedures.</li> </ul>	<ul style="list-style-type: none"> <li>• apply a range of simple problem-solving strategies to predictable challenges.</li> <li>• select from a limited set of conventions and apply these in predictable tasks.</li> <li>• select from a range of resources and procedures and apply these in predictable situations.</li> </ul>	<ul style="list-style-type: none"> <li>• apply problem-solving strategies that include partitioning a problem, identifying and working on the related problems or parts.</li> <li>• select from and apply a range of conventions and apply a given set of industry standards, including developer documentation.</li> <li>• consider alternatives when selecting and using resources and procedures.</li> </ul>	<ul style="list-style-type: none"> <li>• apply problem-solving strategies that include clarifying and refining the solution and generalising from one problem situation to another.</li> <li>• apply conventions, justifying their selection, and industry standards that include developer and user documentation.</li> <li>• justify the selection of appropriate resources and procedures, adjusting plans where required.</li> </ul>	<ul style="list-style-type: none"> <li>• apply a wide repertoire of complex problem-solving strategies that include rethinking problem conditions and constraints.</li> <li>• apply appropriate current industry standards demonstrating an understanding of the diversity of these conventions and standards.</li> <li>• develop criteria for selecting and managing resources and procedures and make provisions for changes.</li> </ul>

## Outcome progressions

<b>Outcome 4: Computer-based systems in society</b> Students understand the interrelations between the development and use of computer-based systems, the individual and society.						
	<b>Level 3</b>	<b>Level 4</b>	<b>Level 5</b>	<b>Level 6</b>	<b>Level 7</b>	<b>Level 8</b>
	Students understand that developers and users participate in the design and development of computer-based systems and that this technology is used in communities.	Students understand that developers and users can influence the design and development of computer-based systems and that this technology has influenced communities.	Students understand how the design and development of computer-based systems is influenced by developers and users and how this technology impacts on individuals and communities.	Students understand how developers and users interact with the design, development and use of computer-based systems and how society and these computer-based systems interact.	Students understand that the relationships between society, developers, users and the design and development of computer-based systems are based on many interconnected factors.	Students understand the complexity of the dynamic relationships between society, developers and users and the design, development and use of computer-based systems.
<b>Students:</b>						
<ul style="list-style-type: none"> <li>understand that developers' attitudes and values affect the development of computer-based systems.</li> <li>understand that users' attitudes and values affect the development and use of computer-based systems.</li> <li>understand there are legal, societal and ethical impacts when computer-based systems are developed and adopted.</li> </ul>	<ul style="list-style-type: none"> <li>understand that developers have attitudes and values that can influence the design and development of computer-based systems.</li> <li>understand that users have attitudes and values that can influence the design, development and use of computer-based systems.</li> <li>understand there are local laws and community norms to consider when designing and developing computer-based systems.</li> </ul>	<ul style="list-style-type: none"> <li>understand the importance of developers' values and attitudes in influencing the design and development of computer-based systems.</li> <li>understand that the needs, values and attitudes of users influence the design, development and use of computer-based systems.</li> <li>understand the importance of local laws and community norms when designing and developing computer-based systems and their resultant benefits and consequences.</li> </ul>	<ul style="list-style-type: none"> <li>understand how developers can have a range of values, attitudes and beliefs that impact on the development of computer-based systems in a variety of ways.</li> <li>understand that users have a range of values, attitudes and beliefs that impact on the design, development and use of computer-based systems in a variety of ways.</li> <li>understand the importance of adherence to local laws and community norms and the impacts these technologies have on individuals, the community and environment.</li> </ul>	<ul style="list-style-type: none"> <li>understand how the developers' awareness of the importance of others' needs, values and attitudes influences the design and development of computer-based systems.</li> <li>understand the key values, attitudes and abilities of users that impact on computer-based systems and how these factors influence the design, development and use of computer-based systems.</li> <li>understand the legal, societal and ethical requirements of computer-based systems and the balance between benefits and costs to individuals, communities and environments.</li> </ul>	<ul style="list-style-type: none"> <li>understand the influence that resource availability, changing circumstances and others' needs and attitudes have on developers and these impacts on the design and development of a computer-based system.</li> <li>understand the range of interconnected beliefs, values, abilities and ethical positions of users, and the range of impacts on the design, development and use of computer-based systems.</li> <li>understand that changing circumstances influence the use of computer-based systems in different communities and how computer-based solutions can be designed to adhere to the social, cultural and legislative requirements.</li> </ul>	<ul style="list-style-type: none"> <li>understand how the developers' awareness of accountability, social responsibilities and the underlying values of all stakeholders influences the design and development of computer-based systems.</li> <li>understand the complex interrelationship between the changing needs and values of users, the community and systems, in the design, development and use of quality computer-based systems.</li> <li>understand how computer-based solutions can adhere to the complex social, cultural and legislative requirements of all stakeholders and the sustainability and implications for future computer-based systems.</li> </ul>