



SAMPLE COURSE OUTLINE

COMPUTER SCIENCE
ATAR YEAR 11

Copyright

© School Curriculum and Standards Authority, 2014

This document – apart from any third party copyright material contained in it – may be freely copied, or communicated on an intranet, for non-commercial purposes in educational institutions, provided that the School Curriculum and Standards Authority is acknowledged as the copyright owner, and that the Authority's moral rights are not infringed.

Copying or communication for any other purpose can be done only within the terms of the *Copyright Act 1968* or with prior written permission of the School Curriculum and Standards Authority. Copying or communication of any third party copyright material can be done only within the terms of the *Copyright Act 1968* or with permission of the copyright owners.

Any content in this document that has been derived from the Australian Curriculum may be used under the terms of the [Creative Commons Attribution-NonCommercial 3.0 Australia licence](#)

Disclaimer

Any resources such as texts, websites and so on that may be referred to in this document are provided as examples of resources that teachers can use to support their learning programs. Their inclusion does not imply that they are mandatory or that they are the only resources relevant to the course.

Sample course outline

Computer Science – ATAR Year 11

Semester 1 – Unit 1

Week	Key teaching points	
	Knowledge	Skills
1	<p>Introduction</p> <ul style="list-style-type: none"> overview of semester 1 assessment requirements <p>Systems analysis and development</p> <ul style="list-style-type: none"> concept of project management, including: <ul style="list-style-type: none"> planning scheduling budgeting tracking types of system development methodologies <ul style="list-style-type: none"> prototyping system development life cycle (SDLC) 	
2–3	<p>Systems analysis and development</p> <ul style="list-style-type: none"> stages of the SDLC <ul style="list-style-type: none"> preliminary analysis analysis design development implementation evaluation and maintenance systems development documentation as a part of the SDLC <ul style="list-style-type: none"> context diagrams using Yourdon/DeMarco notation data flow diagrams (DFD) using Yourdon/DeMarco notation 	<p>Systems analysis and development</p> <ul style="list-style-type: none"> analyse context diagrams and data flow diagrams apply context and data flow diagrams using Yourdon/DeMarco notation, as a part of the SDLC <ul style="list-style-type: none"> create context diagrams create Level 0 DFD's (maximum four processes), applying correct symbols and rules and defining system boundaries
4–5	<p>Systems analysis and development</p> <ul style="list-style-type: none"> types of primary and secondary storage, including: <ul style="list-style-type: none"> random access memory (RAM) read only memory (ROM) cache mechanical disk solid state drive (SSD) concept of boot process (power up to OS booting, including POST) storage capacities, including: <ul style="list-style-type: none"> bit byte kilobyte megabyte gigabyte terabyte hardware and software components for a computer system designed for a specific purpose, including: <ul style="list-style-type: none"> input 	

Week	Key teaching points	
	Knowledge	Skills
	<ul style="list-style-type: none"> ▪ output ▪ processing ▪ storage (primary and secondary) • role of standard operating environment (SOE) • role of components in the central processing unit (CPU) <ul style="list-style-type: none"> ▪ arithmetic logic unit (ALU) ▪ control unit (CU) ▪ registers ▪ program counter ▪ system clock ▪ data, address and control bus • concept of the fetch-execute cycle • troubleshooting strategies, including: <ul style="list-style-type: none"> ▪ diagnosis of fault ▪ implement a solution ▪ document troubleshooting procedure • appropriate physical preventative maintenance measures • purpose of an ICT code of conduct • ethics in the development and use of ICT systems • piracy considerations in the development and use of ICT systems • digital communications etiquette when using ICT systems 	
6–8	<p>Managing data</p> <ul style="list-style-type: none"> • spreadsheet terms, including: <ul style="list-style-type: none"> ▪ cell ▪ formula ▪ label ▪ functions (sum, average, max, min, count, countif) ▪ worksheet ▪ lookup tables (hlookup, vlookup) 	<p>Managing data</p> <ul style="list-style-type: none"> • create solutions using a spreadsheet application using: <ul style="list-style-type: none"> ▪ functions ▪ charts ▪ lookup table ▪ sorting
9	<p>Managing data</p> <ul style="list-style-type: none"> • hierarchical structure of data <ul style="list-style-type: none"> ▪ character/byte ▪ field/attribute ▪ record/tuple ▪ table/entity/relation • data protection methods, including: <ul style="list-style-type: none"> ▪ encryption <ul style="list-style-type: none"> ○ private key ○ public key ▪ authentication <ul style="list-style-type: none"> ○ passwords ○ biometrics ○ digital signature 	

Week	Key teaching points	
	Knowledge	Skills
10–14	<p>Managing data</p> <ul style="list-style-type: none"> • data types, including: <ul style="list-style-type: none"> ▪ number ▪ date/time ▪ currency ▪ text (string) ▪ Boolean (true/false) • database terms <ul style="list-style-type: none"> ▪ data, field, record, relation, atomicity ▪ data integrity ▪ data redundancy • ethical and legal issues relating to the personal use and storage of data • legal requirement and implication of information kept by various organisations about individuals • design considerations for visual interfaces and navigation systems within database systems • purpose of database documentation for the user • data modelling using Chen’s notation entity relationships diagrams 	<p>Managing data</p> <ul style="list-style-type: none"> • resolve simple many to many (M:N) relationship in a multi-table relational database system (maximum three entities) • create using Chen’s notation entity relationship (ER) diagrams for a simple database solution (maximum three entities) • create a working relational multi-table database using: <ul style="list-style-type: none"> ▪ data types ▪ relations ▪ primary and foreign keys ▪ relationships ▪ cardinality (1:1, 1:M, M:1, M:N) ▪ validation rules ▪ forms ▪ reports ▪ queries • create a visual interface for a database • create database documentation
15	Revision	
16	Semester 1 examination	

Semester 2 – Unit 2

Week	Key teaching points	
	Knowledge	Skills
1	<p>Course review</p> <ul style="list-style-type: none"> • review unit 1 • review assessment requirements • overview of semester 2 <p>Developing software</p> <ul style="list-style-type: none"> • evolution of programming languages <ul style="list-style-type: none"> ▪ machine ▪ assembler ▪ procedural ▪ non-procedural ▪ object-oriented • purpose and function of software to operate a computer system <ul style="list-style-type: none"> ▪ operating systems ▪ utility software <ul style="list-style-type: none"> ○ file compression ○ defragmenter ○ anti-virus ○ anti-malware ▪ application software • requirements for software licensing, including: <ul style="list-style-type: none"> ▪ freeware ▪ open source ▪ shareware 	
2–3	<p>Developing software</p> <ul style="list-style-type: none"> • stages of the software development cycle (SDC) <ul style="list-style-type: none"> ▪ state the problem ▪ plan and design ▪ develop ▪ test ▪ evaluate • factors affecting the development of software, including: <ul style="list-style-type: none"> ▪ user needs ▪ user interface 	<p>Developing software</p> <ul style="list-style-type: none"> • create a system solution using the SDC • apply software development requirements
4–6	<p>Programming</p> <ul style="list-style-type: none"> • characteristics of data types <ul style="list-style-type: none"> ▪ integer ▪ real (floating point number) ▪ Boolean ▪ character • appropriate naming conventions for variables • types of code, including: <ul style="list-style-type: none"> ▪ source ▪ executable • types of control structures, including: <ul style="list-style-type: none"> ▪ sequence ▪ selection <ul style="list-style-type: none"> ○ one-way (if then) ○ two-way (if then else) ○ multi-way (case, nested if) 	<p>Programming</p> <ul style="list-style-type: none"> • create flow charts to represent a programming solution • use pseudocode to represent a programming solution

Week	Key teaching points	
	Knowledge	Skills
	<ul style="list-style-type: none"> ▪ iteration <ul style="list-style-type: none"> ○ test first (while) ○ test last (repeat until) ○ fixed (for) • types of program or code errors, including: <ul style="list-style-type: none"> ▪ syntax errors ▪ run-time errors ▪ logical errors • modelling of an algorithm using trace tables to test for logic • purpose of internal and external documentation • modelling of an algorithm using flow charts 	
7–11		<p>Programming</p> <ul style="list-style-type: none"> • apply, using pseudocode and a programming language, the following programming concepts: <ul style="list-style-type: none"> ▪ constants ▪ variables • apply, using pseudocode and a programming language, the following control structures: <ul style="list-style-type: none"> ▪ sequence ▪ selection <ul style="list-style-type: none"> ○ one-way (if then) ○ two-way (if then else) ○ multi-way (case, nested if) ▪ iteration <ul style="list-style-type: none"> ○ test first (while) ○ test last (repeat until) ○ fixed (for) • apply, using pseudocode and a programming language, the following techniques: <ul style="list-style-type: none"> ▪ develop internal and external documentation ▪ select and apply suitable test data for checking the solution ▪ use trace tables to test for and debug logic errors • apply the SDC to create digital solutions • use of the following number systems within a computer <ul style="list-style-type: none"> ▪ binary ▪ decimal ▪ hexadecimal

Week	Key teaching points	
	Knowledge	Skills
12–13	<p>Networks and communications</p> <ul style="list-style-type: none"> functions of the following computer hardware components required for industry networks: <ul style="list-style-type: none"> router switch firewall modem network interface card (NIC) wireless access point (WAP) bridge types of communication networks <ul style="list-style-type: none"> personal area network (PAN) local area network (LAN) wide area network (WAN) worldwide interoperability for microwave access (WiMAX) wireless (PAN, LAN, WAN) technologies appropriate for the implementation of a client/server and peer-to-peer network star network topology diagrammatic representation of network topologies for LAN and WAN characteristics of transmission media, including: <ul style="list-style-type: none"> twisted pair (unshielded twisted pair [UTP] and shielded twisted pair [STP]) fibre optic satellite microwave cellular wireless 	<p>Networks and communications</p> <ul style="list-style-type: none"> create network diagrams using CISCO network diagram conventions to represent network topologies for LAN and WAN
14	<p>Networks and communications</p> <ul style="list-style-type: none"> communication terms <ul style="list-style-type: none"> protocols digital analogue ethernet bandwidth types of communication protocols, including: <ul style="list-style-type: none"> file transfer protocol (FTP) hypertext transfer protocol (HTTP) hypertext transfer protocol secure (HTTPS) simple mail transfer protocol (SMTP) wireless access protocol (WAP) methods used to ensure security of information over the internet, including: <ul style="list-style-type: none"> authentication encryption firewalls types of malware, including: <ul style="list-style-type: none"> viruses worms trojans spyware 	

Week	Key teaching points	
	Knowledge	Skills
15	Revision	
16	Semester 2 examination	