Government of **Western Australia**
School Curriculum and Standards Authority

SAMPLE COURSE OUTLINE

COMPUTER SCIENCE

ATAR YEAR 12

Acknowledgement of Country

Kaya. The School Curriculum and Standards Authority (the Authority) acknowledges that our offices are on Whadjuk Noongar boodjar and that we deliver our services on the country of many traditional custodians and language groups throughout Western Australia. The Authority acknowledges the traditional custodians throughout Western Australia and their continuing connection to land, waters and community. We offer our respect to Elders past and present.

Disclaimer

Any resources, such as texts and websites, 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 mandated or that they are the only resources relevant to the course. Teachers must exercise their professional judgement as to the appropriateness of any resources they may wish to use.

Sample course outline

Computer Science – ATAR Year 12

Unit 3: Design and development of programming and networking solutions

Week	Syllabus content	
	Knowledge	**Skills**

1	**Course introduction**
	• overview of Unit 3
	• assessment requirements
	
	Programming
	Good programming practice
	Framework for development
	• investigate
	  ▪ problem description
	  ▪ define requirements
	  ▪ development schedule (including Gantt charts)
	• design
	  ▪ design data structures
	  ▪ design and test algorithm
	• develop
	  ▪ develop and debug code
	  ▪ unit testing
	• evaluate
	  ▪ user acceptance testing
	  ▪ developer retrospective
	• compare common development processes
	  ▪ linear
	  ▪ iterative
	• good programming practice, including:
	  ▪ validate input before processing
	  ▪ a clear and uncluttered mainline
	  ▪ one logical task per subroutine
	  ▪ use of stubs
	  ▪ appropriate use of control structures and data structures
	  ▪ writing for subsequent maintenance
	  ▪ version control
	  ▪ regular backup
	  ▪ recognition of relevant social and ethical issue
	  ▪ exception handling
	  ▪ functions return a single data structure or value	

Week	Syllabus content	
	Knowledge	**Skills**

2–3	**Programming skills and concepts**
	• program control structures, including:
	  ▪ sequence
	  ▪ selection
	  ▪ iteration
	  ○ fixed
	  ○ post-test
	  ○ pre-test
	• characteristics of data types used in solutions, including:
	  ▪ integer
	  ▪ float
	  ▪ string
	  ▪ Boolean
	• modular coding using functions, parameters and arguments
	  ▪ scope of variables (Global, Local)
	  ▪ parameter passing (value and reference)
	• characteristics of the following data structures:
	  ▪ arrays
	  ○ one-dimensional arrays
	  ○ two-dimensional arrays
	  ▪ dictionaries
		Programming skills and concepts
		• apply, using pseudocode and/or the Python programming language, program control structures in solutions
		• apply, using pseudocode and/or the Python programming language, data types used in solutions as variables
		• apply, using pseudocode and/or the Python programming language, modular coding using functions, parameters and arguments
		• use different types of operators
		  ▪ arithmetic operators (+, -, *, /, %)
		  ▪ relational operators (==, !=, >, =, <=)
		  ▪ logical operators (AND, OR, NOT)
		• reading and writing of complex logical expressions, including:
		  ▪ Boolean operators (AND, OR, NOT)
		  ▪ logical order of precedence
		• apply, using pseudocode, the following data structures:
		  ▪ arrays:
		  ○ one-dimensional arrays
		  ○ two-dimensional arrays
		• dictionaries
		
		Good programming practice
		• apply common development processes
		• apply good programming practice when developing a software solution, including:
		  ▪ validate input before processing
		  ▪ a clear and uncluttered mainline
		  ▪ one logical task per subroutine
		  ▪ use of stubs
		  ▪ appropriate use of control structures and data structures
		  ▪ writing for subsequent maintenance
		  ▪ version control
		  ▪ regular backup
		  ▪ recognition of relevant social and ethical issues
		  ▪ exception handling
		  ▪ functions return a single data structure or value

Week	Syllabus content	
	Knowledge	**Skills**

4–5	**Structured algorithms**
	• effective structure of code
	  ▪ use of a modular approach, including functions and classes
	  ▪ use of stubs to represent incomplete modules
	  ▪ parameter passing
	• common algorithms
	  ▪ purpose of Big O notation
	  ▪ identify Big O notation for search and sort algorithms
	  ▪ search algorithms
	  ○ linear search
	  ○ binary search
	  ▪ sort algorithms
	  ○ bubble sort
	  ○ insertion sort
	  ○ selection sort
		Structured algorithms
		• use pseudocode and/or Python programming language for representing algorithms
		• create code with effective structure
		  ▪ use of a modular approach, including functions and classes
		  ▪ parameter passing
		  ▪ use of stubs to represent incomplete functions

6–7	**Testing**
	• acceptance testing of functional requirements based on user needs
	• the use of live test data to ensure that testing accurately reflects the expected environment in which the new system will operate
	  ▪ large file sizes
	  ▪ mix of transaction types
	  ▪ response times
	• volume of data (load testing)
	• validation of the solution with the design specifications
	• generating relevant test data for complex solutions
	• comparison of actual with expected output
	• unit tests to ensure code performs correctly
	
	Error detection and debugging code
	• the process of detecting and correcting errors, including:
	  ▪ types of error
	  ○ syntax errors
	  ○ logic errors
	  ○ runtime errors, including:
	  – division by zero
	  – index out of range
		Testing
		• comparison of the solution with the design specifications
		• generating relevant test data for complex solutions
		• comparison of actual with expected output
		
		Error detection and debugging code
		• detecting and correcting errors, including:
		  ▪ types of error
		  ▪ syntax errors
		  ▪ logic errors
		  ▪ runtime errors, including:
		  ○ division by zero
		  ○ index out of range
		• methodical approach to the isolation of errors
		  ▪ use of debugging techniques
		  ▪ desk checking (trace tables)
		  ▪ comparison of actual with expected output

Week	Syllabus content	
	Knowledge	**Skills**

	• methods of error detection and correction
	  ▪ methodical approach to the isolation of logic errors
	  ▪ use of debugging techniques
	  ○ breakpoints
	  ○ print statements
	  ▪ desk checking (trace tables)
	  ▪ comparison of actual with expected output	

8	**Object-oriented programming**
	• characteristics of the following object-oriented concepts:
	  ▪ classes
	  ▪ objects
	  ▪ attributes
	  ▪ methods
	  ▪ abstraction and polymorphism
	  ▪ instantiation
	  ▪ inheritance
	  ▪ encapsulation
		Object-oriented programming
		• apply, using pseudocode and/or the Python programming language, object-oriented programming concepts, including:
		  ▪ classes
		  ▪ objects
		  ▪ attributes
		  ▪ methods/operations
		  ▪ abstraction
		  ▪ instantiation
		  ▪ inheritance
		• use an object-oriented programming language to:
		  ▪ create classes and objects with attributes and methods
		  ▪ instantiate objects
		  ▪ use inheritance to extend classes
		  ▪ use variables and control structures within objects

9–10	**Ethical and legal implications for developers of software**
	• impacts of software in society, including:
	  ▪ computer malware, such as viruses
	  ▪ reliance on software
	  ▪ social networking
	  ▪ cyber safety
	  ▪ large volumes of information (which may be unsupported, unverifiable, misleading or incorrect) available through the internet
	• rights and responsibilities of software developers
	  ▪ acknowledging the intellectual property of others
	  ▪ recognition by others of the developer's intellectual property	

Week	Syllabus content	
	Knowledge	**Skills**
	▪ producing quality software solutions ▪ appropriately responding to user-identified problems ▪ adhering to code of conduct ▪ neither generating nor transmitting malware ▪ addressing ergonomic issues in software design ▪ ensuring software addresses inclusivity issues ▪ ensuring individuals' privacy is not compromised	
11–12	**Networking communications** **Models of networking** • comparison between the Open Systems Interconnection (OSI) Model and the Department of Defence (DoD) transmission control protocol/internet protocol (TCP/IP) model • OSI model: ▪ purpose of OSI model ▪ layers of OSI model ▪ role of layers within the model ▪ MAC address layer 2 switching ▪ IP address layer 3 routing • DoD TCP/IP model: ▪ purpose of DoD model ▪ layers of DoD model ○ application ○ transport ○ internet ○ network ▪ role of layers within the model • key protocols and devices associated with layers in models ▪ user datagram protocol (UDP) ▪ difference between UDP and TCP ▪ compare packet architecture between TCP and UDP • features of TCP/IP • IPv4 vs IPv6 • IP Addressing • Private vs public IP addressing • subnetting ▪ subnet masks	**Models of networking** • identify, design and apply IP addressing scheme • identify and design subnets, applying subnet masks and CIDR notation

Week	Syllabus content	
	Knowledge	**Skills**
	▪ Classless Inter-Domain Routing (CIDR) notation • default gateways • domain name system (DNS) • ports • packet architecture ▪ data ▪ segment ▪ packet ▪ frame ▪ bits	
13–14	**Network components** • role of components at different network layers ▪ transmission media ▪ modem ▪ router ▪ gateway ▪ switch ▪ wireless access point ▪ firewalls **Network performance** • network design topology for security and performance • bandwidth ▪ mapping networks using diagrams including intermediary and end devices ▪ subnetting and broadcast domains (segmentation) • tools for network performance management and troubleshooting ▪ ping ▪ traceroute	**Network performance** • use ping and traceroute to troubleshoot and evaluate network performance • interpret and create network diagrams using specified CISCO conventions to represent network topologies, considering addressing, subnets, segmentation, security and performance
15	Revision	
16	Semester 1 examination	

Unit 4: Design and development of database solutions and cybersecurity considerations

Week	Syllabus content	
	Knowledge	**Skills**
1–2	**Course introduction** • review of Unit 3 • overview of Unit 4 • assessment requirements **Cybersecurity** **Network threats** • external network threats ▪ social engineering (phishing) ▪ denial of service, including distributed denial of service ▪ back door ▪ IP spoofing ▪ SQL Injection ▪ man-in-the-middle ▪ cross-site scripting ▪ types of malware ○ viruses ○ worms ○ trojan horses ○ spyware ○ adware ○ ransomware ▪ physical network threats ▪ zero day vulnerabilities • internal network threats ▪ lost or stolen devices ▪ compromised credentials ▪ misuse by employees • security solutions to network threats ▪ analysis of log files ▪ anti-malware ▪ firewall filtering ▪ access control lists ▪ intrusion prevention systems ▪ virtual private networks ▪ user training ▪ ICT code of conduct ▪ physical security • appropriate solutions to different external network threats	**Network threats** • identify network security threats and suggest solutions for a given stimulus (mitigation strategies) **Security frameworks** **Skills** • use the CIA triad to analyse security threats and incidents • use the AAA framework for security analysis and auditing

Week	Syllabus content	
	Knowledge	**Skills**

	Security frameworks
	Knowledge
	• the CIA triad model of security analysis
	  ▪ confidentiality
	  ▪ integrity
	  ▪ availability
	• the AAA framework for securing systems
	  ▪ authentication
	  ▪ authorisation
	  ▪ accounting	

3	**Cryptography**
	• symmetric encryption
	• asymmetric encryption (public/private keys)
	  ▪ certificate purpose and use
	• use of asymmetric encryption to
	  ▪ prevent unauthorised access to data
	  ▪ authenticate data being sent across network
	• secure communication over the internet:
	  ▪ use of asymmetric encryption to establish connection and symmetric encryption to exchange data
	• common methods of encryption
	  ▪ early methods and weaknesses
	  ▪ current best practice
	
	Cybersecurity
	Ethics and Law
	• the *Privacy Act 1988* as it relates to data security Australian Privacy Principle 11 (APP11)
	• the *Privacy Amendment (Notifiable Data Breaches) Act 2017*
		Ethics and Law
		• apply theoretical ethics and law knowledge using supplied case studies

4–5	**Ethical hacking**
	• role of ethical hacking in improving network security
	• penetration testing: role of blue team vs red team
	
	Data management
	Core concepts
	• organisation of a relational database:
	  ▪ entities
	  ▪ attributes	

Week	Syllabus content	
	Knowledge	**Skills**

	▪ relationships
	  ○ one-to-one
	  ○ one-to-many
	  ○ many-to-many
	• tables as the implementation of entities, consisting of fields and records
	• data types
	  ▪ integer
	  ▪ float
	  ▪ Boolean
	  ▪ text
	  ▪ date
	• primary and foreign keys to link tables
	• composite key
	• data anomalies
	  ▪ insert
	  ▪ update
	  ▪ delete
	• how databases interact with other systems and link to the programming content
	  ▪ open database connectivity
	• role of ACID in database transactions – atomicity, consistency, isolation, durability	

6–8	**Data modelling**
	• purpose of database documentation for the developers
	  ▪ data dictionary
	  ▪ entity relationship (ER) diagrams using crow's foot notation
	
	Data integrity
	• data integrity
	  ▪ referential integrity
	  ▪ domain integrity
	  ▪ entity integrity
	• factors influencing quality of data, including:
	  ▪ currency
	  ▪ authenticity
	  ▪ relevance
	  ▪ accuracy
	  ▪ outliers (cleaning)
		Data modelling
		• analyse ER diagrams written in crow's foot notation (up to 10 tables)
		• create accurate ER diagrams using crow's foot notation (three to eight tables)
		• represent databases using relational notation
		• create data dictionaries
		• resolve many to many (M:N) relationships

Week	Syllabus content	
	Knowledge	**Skills**

9–10	**Normalisation**
	• purpose of normalising data to third normal form (3NF)
	  ▪ rules of 1NF
	  ▪ rules of 2NF
	  ▪ rules of 3NF
	• know the process to normalise data to 3NF
		Normalisation
		• apply the process to normalise data to 3NF using relational notation

11–13	**Database creation and manipulation**
	• know techniques to retrieve required information through querying data sets using SQL
	• purpose of cleaning and manipulating data sets to import required data into a database
		Database creation and manipulation
		• use a relational database management system (RDBMS) to create and manipulate a relational database with a minimum of five tables
		• use SQL to create, modify and query a database including:
		  ▪ CREATE and MODIFY tables, including:
		  ○ use of constraints to ensure validity of data
		  ○ enforcing referential integrity
		  ▪ DROP
		  ▪ SELECT
		  ▪ INSERT
		  ▪ UPDATE
		  ▪ DELETE, including cascading deletes
		  ▪ aggregate functions (COUNT, SUM, AVG, MAX, MIN)
		  ▪ ORDER BY
		  ▪ inner join across multiple tables (up to 4 tables)
		  ▪ GROUP BY (with aggregate functions)
		  ▪ calculated fields
		  ▪ concatenated fields
		  ▪ use of aliases with calculated and concatenated fields
		• clean and manipulate data sets to import required data into a database

14	**Development issues**
	• ethical issues
	  ▪ collecting data about individuals
	  ▪ privacy concerns
	  ▪ appropriate use of data
	  ▪ reliability of data sources
	  ▪ acknowledgement of data sources
	  ▪ use of data mining
	• security issues
	  ▪ keeping personal data private	

Week	Syllabus content	
	Knowledge	**Skills**
	▪ backups of organisational data ▪ restricting access to data ▪ ownership and control of data • legal issues ▪ Australian privacy law in relation to database development ▪ Australian Privacy Principles (APP5, APP8, APP10, APP11, APP12)	
15	Revision	
16	Semester 2 examination	