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

COMPUTER SCIENCE

General course

Year 11 syllabus for teaching from 2026

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.

Important information

As part of the Western Australian Certificate of Education (WACE) Refreshment, the School Curriculum and Standards Authority (the Authority) has revised the course rationale and aims, and updated the General Capabilities to create clearer connections with the syllabus content.

This syllabus is effective from 1 January 2026.

Users of this syllabus are responsible for checking its currency.

Syllabuses are formally reviewed by the Authority on a cyclical basis, typically every five years.

Contents

Rationale

The Computer Science General course focuses on the fundamental principles, concepts and skills of the technology field, and provides students with opportunities to develop flexibility and adaptability in their application as developers and users. The course explores the underpinning knowledge and skills of computer science and how these are applied in a practical way to the development of computer systems and software, as well as the connectivity between computers, peripheral devices and software used in the home and in the workplace.

The course investigates personal and social impacts of technological developments. It explores the ethical and legal factors that influence developments in information technology so that students can identify the consequences of decisions made by developers. Students are given the opportunity to develop and maintain databases and digital solutions using a programming language and skills.

Throughout the course, students develop problem-solving and technical skills as they learn how to diagnose and solve problems, improving their understanding of the building blocks of computing. They also develop skills in computational thinking, programming and software development, data management and security, digital systems and networking, and collaboration and project management.

This course provides students with practical and technical skills that equip them to function effectively in daily life and are vital for employability, including programming, data analysis, cybersecurity awareness, project management and problem-solving through a systematic approach. It provides a sound understanding of computing to support students pursuing further studies in related fields.

The Computer Science General course provides students with foundational skills that prepare them for further education, vocational training and employment in various technology-related fields. Students may develop skills to support employment opportunities in the information technology field as a support technician, junior web developer, helpdesk operator or network assistant, or in data entry or administration.

Aims

The Computer Science General course aims to develop students':

- skills in designing, maintaining, adapting and producing databases and digital solutions
- problem-solving skills using algorithms, data structures and programming languages
- skills in assessing networking issues within a digital environment
- understanding of the design, application and interactions of data and software in digital systems through the creation and maintenance of databases, network data transmission and programming constructs
- understanding of how to apply a technology process accurately to develop a digital solution
- understanding of the interrelationships between the development and use of digital solutions for individuals and societies in relation to the legal and ethical implications of software design and data management.

Organisation

This course is organised into a Year 11 syllabus and a Year 12 syllabus. The cognitive complexity of the syllabus content increases from Year 11 to Year 12.

Structure of the syllabus

The Year 11 syllabus is divided into two units, each of one semester duration, which are typically delivered as a pair. The notional time for each unit is 55 class contact hours.

Unit 1 – Personal use of computer systems

This unit provides students with the knowledge and skills required to use and maintain a personal computer. It introduces a formal method for developing simple information systems and databases. While considering personal needs, students examine the social, ethical and legal implications of personal computer use.

Unit 2 – Personal use of communication and information systems

This unit introduces a formal method for developing networks and internet technologies and writing a sequence of simple instructions. Students examine the social, ethical and legal implications associated with software development.

Each unit includes:

- a unit description – a short description of the focus of the unit
- unit content – the content to be taught and learned.

Organisation of content

The unit content includes both theoretical aspects (Knowledge) and practical aspects (Skills).

The course is divided into five content areas.

Unit 1 is divided into two content areas:

- Systems analysis and development
- Managing data.

Unit 2 is divided into three content areas:

- Developing software
- Programming
- Networks and communications.

Systems analysis and development

The functions and technical capabilities of systems, how components are configured to form a computer system, and factors which affect the design of an information system, are explored. The compatibility of components, output, bandwidth considerations, and usability, security, health and safety considerations are explored. Evaluations of systems, devices or components are conducted, while acquiring computer hardware knowledge and skills.

Managing data

The distinction between data and information, including the different types of data (including text and number) and the varied representation of data within a computer, are addressed. The representation of data types, the graphical representation of data, and how data is stored using a database are also addressed.

Developing software

A systems development cycle (SDC) that includes some basic systems engineering and the application of standards is applied. How a developer's interactions with users affect the development and use of the system is investigated. Various methods of developing software systems and the problems associated with connecting systems in an increasingly global environment are addressed. The different perspectives of users and developers to the development and use of computer-based systems are explored.

Programming

The different types of programming languages are investigated. The basic constructs of sequence, selection and iteration are examined. The analysing and breaking up problems into small, self-contained units for which procedures or functions are created in a programming language are addressed. The passing of parameters to procedures, functions and modules are explored.

Networks and communications

The various structures and components of a network, including the communication media used to combine them, are examined. The convergence of technologies which involves the integration of computers and communication hardware, is investigated. Similarly, the design and creation of networks of various configurations, as well as connecting networks of different types, are investigated. The application of connectivity standards relating to networks and the internet, is addressed. Communication software models and standards; the types, purpose and use of protocols, servers and operating systems in communications; and software and the aspects to consider in network security are explored.

Resources

It is recommended that for delivery of the Computer Science General course, students have access to the following resources:

- computers with access to the internet
- peripheral devices, including
  - scanner/photocopier/printer (multi-function device)
  - printer/s
- applications software
  - spreadsheet software
  - word processing software
  - presentation software
  - multimedia software
  - personal communication software
  - collaborative management software
  - browser software
  - web-authoring software.

Programming language

There is no prescribed programming language for the Computer Science General course. However, to meet the assessment requirements for this syllabus, it is required that students use a programming language that enables the:

- development of a purpose-designed software solution
- design, creation, modification, testing, evaluation and documentation of programs
- writing, compiling, interpretation, testing and debugging of code
- use and development of a user interface.

For the Computer Science General syllabus, the programming language should provide the student with opportunity to:

- use control structures, including sequence, selection and iteration
- construct and use data structures
- design and implement data validation techniques
- apply modularised and structured programming methods using modularisation and parameter passing.

There is no requirement within the Computer Science General course to create a user interface, unless required for a particular programming language (e.g. PHP).

The suggested programming languages for the Computer Science General syllabus are:

- Scratch
- Alice
- macros – VBA, application specific macros, scripting languages, Unix Bash
- Applescript
- Gamefroot
- GameMaker
- GameSalad
- Visual Basic
- programmable robotic software – RCX Code, ROBOLAB, RoboMind.

Database management systems

There is no prescribed database management system for the Computer Science General course. However, to meet the assessment requirements for this syllabus, it is required that students use a database management system that enables the:

- development of a purpose-designed database solution
- design, creation, modification, testing and evaluation of a database solution
- creation of tables, queries, forms and reports
- use and development of a user interface.

The database management systems should provide the student with opportunity to:

- create a working database
- construct simple queries.

The suggested database management system software for the Computer Science General course are:

- Microsoft Access
- MySQL
- FileMaker
- FoxPro
- Paradox
- a spreadsheet application.

Progression from the Years 7–10 curriculum

This syllabus continues to develop student learning around the knowledge, understandings and skills within the Years 7–10 Digital Technologies curriculum and focuses on the components of digital systems (software, hardware and networks) and their uses; the representation of data; and how data are represented and structured symbolically.

The syllabus continues to develop students' skills by producing digital solutions through acquiring, managing and analysing data; defining problems; designing solutions; implementing and evaluating solutions; and managing projects

Representation of the General Capabilities

The General Capabilities encompass the knowledge, skills, behaviours and dispositions that will support students to live and work successfully now and into the future. They are not assessed unless identified within the specified unit content. Teachers should find opportunities to incorporate the following General Capabilities into the teaching and learning program for the Computer Science General course.

Critical and creative thinking

Students develop their critical and creative thinking skills through processes such as inquiring, generating, analysing and reflecting. They ask meaningful questions and explore problems in computing, conducting research to understand different technologies, algorithms and systems. Students engage in generating creative solutions, designing innovative applications, coding new programs and developing digital systems. They evaluate the efficiency and functionality of their solutions, testing algorithms and assessing system performance, and identifying areas for improvement. Students review their work, consider alternative approaches, and refine their coding practices while contemplating the broader ethical, social and environmental implications of technology.

Digital literacy

Students develop digital literacy by incorporating skills such as practising digital safety and wellbeing, investigating, creating and exchanging, and managing and operating. They learn to safeguard personal data, apply security protocols and address topics like cybersecurity and ethical online behaviour. Students research computing systems, technologies and ethical dilemmas in technology, critically analysing digital tools and their impact. They design, develop and test software solutions, communicate their findings through digital platforms, and collaborate. Students configure and troubleshoot digital systems, ensuring the smooth operation of hardware and software, and maintain network security.

Ethical understanding

Students develop their ethical understanding by focusing on ethical issues in the development and use of technology systems. They consider privacy, security and fairness, particularly with regards to how user data is collected, stored and shared. Students explore ethical and legal issues, such as digital communications etiquette, and are encouraged to engage in responsible, respectful communication when using technology systems. They also reflect on the broader social and environmental impacts of technology, ensuring that their solutions are both responsible and sustainable.

Literacy

Students develop literacy skills, particularly in reading, viewing and writing, throughout the course. They engage with technical documentation, such as programming manuals, system specifications and case studies, which they interpret and analyse to demonstrate their comprehension of complex technologies concepts. Students also review and evaluate code, algorithms and system designs, building their ability to understand and apply specialised terminology and symbols. They document their work through projects, where they write clear, structured reports, including code comments, user manuals and technical summaries that explain system functionality, design choices and testing procedures. Students write and modify code, requiring precise communication through documentation and error analysis.

Numeracy

Students develop numeracy skills, particularly in number and algebra, and statistics and probability, by applying mathematical concepts to solving computing problems. They work with numerical data, and binary and hexadecimal number systems, and use mathematical operations in programming for algorithms calculations and conditional logic. Students apply algebraic thinking when using variables, functions and loops in coding. They collect, organise and interpret data to analyse trends, such as measuring system performance, processing datasets and understanding probability in cybersecurity (e.g. encryption strength and risk assessment).

Addressing the other General Capabilities

Although the following General Capabilities have not been identified as a focus in the Computer Science General Year 11 syllabus, teachers may find opportunities to incorporate them into the teaching and learning program.

- Intercultural understanding
- Personal and social capability

Such opportunities may occur through the application of different contexts, pedagogical practices and/or assessment strategies that relate to the syllabus as part of the teaching and learning program.

Summary representation of the General Capabilities in the Computer Science course

The unit content and assessment types for this course provide students with the opportunity to develop the General Capabilities summarised in the table below.

Year	Course	Course type	General Capabilities						
			CCT	DL	EU	IU	L	N	PSC
Year 11	Computer Science (GECSC)	General	✓	✓	✓		✓	✓	
Year 12	Computer Science (GTCSC)	General	✓	✓	✓		✓	✓	

Key
CCT: Critical and creative thinking, DL: Digital literacy, EU: Ethical understanding, IU: Intercultural understanding, L: Literacy, N: Numeracy, PSC: Personal and social capability

Representation of the Cross-curriculum Priorities

The Cross-curriculum Priorities address contemporary issues which students face in a globalised world. Teachers may find opportunities to incorporate them into the teaching and learning program for the Computer Science General course. The Cross-curriculum Priorities are not assessed unless they are identified within the specified unit content.

Aboriginal and Torres Strait Islander histories and cultures

The Computer Science General course may provide opportunities for students to explore creative, engaging and diverse learning contexts for students to value and appreciate the contribution by the world's oldest continuous living cultures to past, present and emerging technologies.

Asia and Australia's engagement with Asia

The Computer Science General course may provide opportunities for students to explore contemporary and emerging technological achievements that the Asia region and Pacific region have made, and continue to make, to global technological advances, including; innovation in hardware and software design and development; the regions' role in outsourcing of information and communication technology (ICT) services; and globalisation. Students could also consider the contribution of Australia's contemporary and emerging technological achievements to the Asia and Pacific Regions.

Sustainability

The Computer Science General course may provide opportunities for students, within authentic contexts, to choose and evaluate digital technologies and information systems with regard to risks and opportunities they present. They may also evaluate the extent to which digital solutions can embrace and promote sustainable practices.

Unit 1 – Personal use of computer systems

Unit description

This unit provides students with the knowledge and skills required to use and maintain a personal computer. It introduces a formal method for developing simple information systems and databases. While considering personal needs, students examine the social, ethical and legal implications of personal computer use.

Unit content

This unit includes the knowledge, understandings and skills described below.

The content includes theoretical aspects (Knowledge) and practical aspects (Skills) and is organised into the following areas:

- Systems analysis and development
- Managing data.

Typically, approximately 60 percent of class time would be allocated for the Managing data content and approximately 40 percent would be allocated for Systems analysis and development content.

Systems analysis and development

Knowledge

- purpose of the systems development life cycle (SDLC)
- stages of the SDLC
  - preliminary analysis
  - analysis
  - design
  - development
  - implementation
  - evaluation and maintenance
- flow of data through an information system
  - input
  - processing
  - storage
  - output
- functions of computer hardware components, including:
  - input
    - keyboard
    - mouse
    - microphone
    - digital camera/web cam
    - scanner
  - processing
    - central processing unit (CPU)
    - control unit (CU)

        - arithmetic logic unit (ALU)
        - registers
    - primary storage
        - random access memory (RAM)
        - read only memory (ROM)
    - secondary storage
        - mechanical drive
        - solid state drive
        - online
    - output
        - monitor
        - printer
        - speaker/headphones
- types of computer systems, including:
    - mobile
    - desktop
    - server
- the role of an operating system
- types of hardware booting processes
    - cold
    - warm
- how user wants influence the choice, use and creation of personal computer systems
- types of basic maintenance strategies and computer protection software, including:
    - defragmentation
    - error check
    - disk clean
    - back up
    - anti-malware
- basic maintenance strategies and techniques to rectify simple computer difficulties, including:
    - diagnosis of fault
    - implementation of a solution
    - description of process

Skills

- apply the following hardware booting processes
    - cold
    - warm
- connect peripheral devices to a computer system using:
    - ports
    - universal serial bus (USB)
    - Firewire
    - PS2
    - ethernet
    - serial

- install simple software
- apply basic care and handling of hardware equipment measures to ensure personal safety and appropriate use of components
- apply basic maintenance strategies and computer protection software
- apply basic maintenance strategies and techniques to rectify simple computer difficulties

Managing Data

Knowledge

- data management techniques for personal computer use, including hierarchical storage of data using files and folders
- issues related to ethics in the storage of personal data
- features of word processing software, including common formatting functions
- features of spreadsheet software, including:
  - simple functions (sum, average, min and max)
  - simple formulae (addition, subtraction, multiplication and division)
- features of database software, including:
  - components of a single table database (field, record, file)
  - data entry forms
  - simple search techniques
  - create a simple query
  - simple data types (number, text, Boolean, date, currency)

Skills

- apply hierarchical file management techniques for personal computer use
- use word processing software
- use spreadsheet software
- use database software

Unit 2 – Personal use of communication and information systems

Unit description

This unit introduces a formal method for developing networks and internet technologies and writing a sequence of simple instructions. Students examine the social, ethical and legal implications associated with software development.

Unit content

This unit builds on the content covered in Unit 1.

This unit includes the knowledge, understandings and skills described below.

The content includes theoretical aspects (Knowledge) and practical aspects (Skills) and is organised into the following areas:

- Developing software
- Programming
- Networks and communications.

Typically, approximately 60 per cent of class time would be allocated for the Programming content, approximately 20 per cent would be allocated for Developing software content, and approximately 20 per cent would be allocated for Networks and communications content.

Developing software

Knowledge

- hardware and software systems used in personal computing (applications, operating systems)
- the roles of hardware, software and the user in a computer system
- interrelationship between users, hardware and software in a personal computer system
- the purpose of the software development cycle (SDC)
- stages of the SDC
  - state the problem
  - plan and design
  - develop the solution
  - test the solution
  - evaluate the solution
- comparison of website construction tools
- requirements for software licensing, including:
  - single user
  - site licence
- ethical and legal issues associated with software, including:
  - copyright
  - piracy

Skills

- modify an existing simple software solution
- develop simple software solutions using the SDC

Programming

Knowledge

- the components of a computer program
  - inputs
  - processing
  - outputs
- control structures
  - sequence
  - selection
  - iteration
- the concepts of variables and data types, including:
  - integer
  - real
  - character
  - string

Skills

- use variables, data types, control structures and a simple programming language to develop a software solution
- use web tools to create linked web pages

Networks and communications

Knowledge

- key concepts, terminology and functions of common network components
  - data transmission rates
    - megabits per second (Mbps)
    - gigabits per second (Gbps)
  - wired data transmission media
    - twisted pair
    - fibre optic
  - wireless transmission
- hardware components required for a personal area network (PAN) or home network, including:
  - modem
  - router
  - wireless access point
  - firewall
- types of communication software, including:
  - browser
  - email

  - web authoring
  - scripting
- software requirements for a PAN or home network, including:
  - browser
  - plugin
  - internet connectivity software
- effect of bandwidth availability on network functionality
- features of a network, including the ability to share:
  - files
  - peripheral devices
  - an internet connection
  - storage devices
- the role of an internet service provider in a PAN or home network
- concept of internet protocols, including:
  - hypertext transfer protocol (HTTP)
  - hypertext transfer protocol secure (HTTPS)
  - file transfer protocol (FTP)
- methods to ensure reliability of internet data for personal use
- measures an individual can take to help maintain data privacy and security
- the role of users in maintaining the security of information transmitted through communication systems

Skills

- connect common peripheral devices
- create and administer a simple peer-to-peer network to:
  - share files
  - share peripheral devices (printer, scanner)
  - share internet connection
- use Bluetooth to create a simple personal network
- use communication software to upload files to a web server
- analyse the suitability of a PAN or a home network solution

School-based assessment

The *Western Australian Certificate of Education (WACE) Manual* contains essential information on principles, policies and procedures for school-based assessment that needs to be read in conjunction with this syllabus.

Teachers design school-based assessment tasks to meet the needs of students. The table below provides details of the assessment types for the Computer Science General Year 11 syllabus and the weighting for each assessment type.

Assessment table – Year 11

Type of assessment	Weighting
Project The student is required to develop a spreadsheet and/or database and/or software system by using the system development life cycle and/or software development cycle. Students are provided with the stimulus materials on which the project is based. Stimulus material can include: diagrams; extracts from newspaper and journal articles; flow charts; trace tables; algorithms and algorithm segments (in pseudocode); and/or screen captures or representations of spreadsheets, databases and programs. Diagrams could include those related to computer system diagrams, and network diagrams. The student is required to research ideas: implement a database and/or software system using a database management system and programming language, to develop and evaluate solutions and manage processes throughout the production.	60%
Theory test Typically include a combination of questions requiring short and extended answers. Short answer questions can be a mix of closed and open items that can be sectionalised or scaffolded. The student can be required to explain concepts, apply knowledge, analyse and/or interpret data and/or respond to stimulus materials. Stimulus material can include: diagrams; extracts from newspaper and journal articles; flow charts; trace tables; algorithms and algorithm segments (in pseudocode); and/or screen captures or representations of spreadsheets, databases and programs. Diagrams could include those related to computer system diagrams, and network diagrams. Extended answer questions can be a mix of closed and open items that can be sectionalised or scaffolded typically with an increasing level of complexity. The student can be required to explain concepts; apply knowledge; analyse and/or interpret data, extended algorithms, databases, spreadsheets, tables and/or diagrams; and/or devise labelled diagrams, solutions (or parts or solutions); and/or respond to stimulus materials. Stimulus material can include: diagrams; extracts from newspaper and journal articles; flow charts; trace tables; algorithms and algorithm segments (in pseudocode); and/or screen captures or representations of spreadsheets, databases and programs. Diagrams could include those related to computer system diagrams, and network diagrams.	20%

Type of assessment	Weighting
Practical test Typically consist of a set of questions requiring the use of spreadsheet software, programming language and/or a database management system. Spreadsheet skills assessed include creating spreadsheets that include formulae and functions Programming skills assessed include: writing code; and/or compiling, testing and/or debugging code. Database skills assessed include: creating fields, data types, keys for tables; queries, forms and/or reports.	20%

Teachers are required to use the assessment table to develop an assessment outline for the pair of units (or for a single unit where only one is being studied).

The assessment outline must:

- include a set of assessment tasks
- include a general description of each task
- indicate the unit content to be assessed
- indicate a weighting for each task and each assessment type
- include the approximate timing of each task (for example, the week the task is conducted, or the issue and submission dates for an extended task).

In the assessment outline for the pair of units, each assessment type must be included at least once over the year/pair of units. In the assessment outline where a single unit is being studied, each assessment type must be included at least once.

The set of assessment tasks must provide a representative sampling of the content for Unit 1 and Unit 2.

Assessment tasks not administered under test/controlled conditions require appropriate validation/authentication processes.

Grading

Schools report student achievement in terms of the following grades.

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

The teacher prepares a ranked list and assigns the student a grade for the pair of units (or for a unit where only one unit is being studied). The grade is based on the student's overall performance as judged by reference to a set of pre-determined standards. These standards are defined by grade descriptions and annotated work samples. The grade descriptions for the Computer Science General Year 11 syllabus are provided in Appendix 1. They can also be accessed, together with annotated work samples, through the Guide to Grades link on the course page of the Authority website at www.scsa.wa.edu.au.

To be assigned a grade, a student must have had the opportunity to complete the education program, including the assessment program (unless the school accepts that there are exceptional and justifiable circumstances).

Refer to the *WACE Manual* for further information about the use of a ranked list in the process of assigning grades.

Appendix 1 – Grade descriptions Year 11

A	**Knowledge and understanding** Uses computer science terminology accurately, and describes processes and concepts in context.
	System development processes Gathers and refines appropriate data from relevant sources for personal use. Conducts an analysis of an information system using an appropriate methodology and provides a detailed and relevant analysis, planning, and presentation of the system requirements. Provides relevant recommendations with justification, reflecting system requirements, and where relevant presents alternatives.
	Data management skills Consistently constructs functional database solutions, with relevant data entry forms and queries, accurately reflecting system requirements. Consistently constructs functional spreadsheet solutions, with relevant functions and formulae, accurately reflecting system requirements.
	Programming skills Designs and effectively applies relevant programming control structures, variables and data types. Creates relevant software solutions accurately reflecting system requirements. Consistently applies the software development cycle effectively to create software solutions accurately reflecting system requirements.

B	**Knowledge and understanding** Uses computer science terminology, and describes processes and concepts in context.
	System development processes Gathers and refines data from relevant sources for personal use. Conducts an analysis of an information system using an appropriate methodology and provides an appropriate analysis, planning and presentation of system requirements. Provides appropriate recommendations, reflecting system requirements, and attempts to justify alternatives.
	Data management skills Consistently constructs functional database solutions, consisting of appropriate data entry forms and queries, reflecting system requirements. Consistently constructs functional spreadsheet solutions, consisting of appropriate functions and formulae, reflecting system requirements.
	Programming skills Designs and effectively applies programming control structures, variables and data types. Creates software solutions accurately reflecting system requirements. Applies the software development cycle effectively to create software solutions, accurately reflecting system requirements.

C	**Knowledge and understanding** Uses computer science terminology, and recalls processes and concepts.
	System development processes Gathers data from a limited range of sources for personal use. Conducts a review of an information system using a methodology and provides an outline, planning and presentation of the system requirements. Provides basic, appropriate recommendations reflecting system requirements drawn from the data collected.
	Data management skills Constructs database solutions, consisting of data entry forms and queries, reflecting system requirements. Constructs spreadsheet solutions, consisting of functions and formulae, reflecting system requirements.
	Programming skills Designs and applies programming control structures, variables and data types. Creates software solutions, occasionally reflecting system requirements. Applies the software development cycle to create software solutions, occasionally reflecting system requirements.

D	**Knowledge and understanding** Attempts to use computer science terminology, and inconsistently recalls processes and/or concepts.
	System development processes Attempts to gather data for personal use. Develops planning and presentation of system requirements which is brief or incomplete. Makes recommendations which are brief and/or inconsistent with the system requirements.
	Data management skills Constructs incomplete database solutions, occasionally reflecting system requirements. Constructs incomplete spreadsheet solutions, occasionally reflecting system requirements.
	Programming skills Unsuccessfully designs and applies, programming control structures variables and data types. Unsuccessfully creates algorithms and/or software solutions. Unsuccessfully applies the software development cycle, to create incomplete software solutions.

E	Does not meet the requirements of a D grade and/or has completed insufficient assessment tasks to be assigned a higher grade.

Appendix 2 – Glossary

This glossary is provided to enable a common understanding of the key terms in this syllabus.

Algorithm
Instructions specifying the logic of a program.

Attribute
A name which defines a field in a database table.

Authentication
A system entry security measure based on user input, such as a digital signature, username and password, or forms of biometrics.

Bandwidth
Architecture: rate of data transfer in bits per second.
Communication: range of frequencies analogue signals can be carried over (measured in hertz).

Benchmarking
Software or hardware performance evaluated against standardised criteria.

Boolean
Database: data type exhibiting only two possible conditions – true or false.
Programming: an expression capable of generating only two possible outcomes – true or false.
Operator: AND, OR, NOT used to combine or exclude keywords in a condition.

Byte code
Object code run on a virtual machine allowing portability across multiple platforms.

Cardinality
The relationship defined between two relational database entities.

Character
Data type, that is one byte in size, able to hold a single alphanumeric entry.

Computer-aided software engineering (CASE)
Software tools use to assist in the development of software during the SDC, including code generation, and the creation of Gantt and PERT charts.

Condition
A statement or expression for which there can only be a true or false outcome.

Constant
Name of a memory location whose literal content does not change during the execution of the program.

Context diagram
Top level diagram that graphically defines the system boundary and the flow of data between the system and external entities.

Control structures
Constructs that control the flow of the program's execution; specifically, sequence, selection and iteration.

Convergence
Process of interlinking different technologies into a single device, for example, smart phones.

Data
Raw facts that represent real-world items which become information when organised. Singular – datum.

Data dictionary
Metadata that describes the attributes of data to be stored in a database.

Data duplication
Data that is physically duplicated across a database.

Data flow diagram (DFD)
Visual representation describing the flow of data through a system.

Data redundancy
Duplication of the same attributes within a table; attribute data that can be derived from other existing data.

Data type
The characteristics of data that can be stored in a cell, such as integer, real, Boolean and string.

Database
A collection of related data which allows input, editing and deletion, and can be queried for patterns, and produce reports and charts.

Documentation
Written text that accompanies software describing attributes, characteristics and/or qualities of the program, including, the code, data dictionary, user manual.

Domain name server (DNS)
The DNS translates host addresses (URL) into Internet Protocol (IP) addresses.

Dynamic host configuration protocol (DHCP)
A protocol that automatically assigns a unique (IP) address and/or subnet mask to a communication device joining a network.

Encryption
Process of encoding data via the implementation of an encryption key.

Entity
An entity is an object, thing person, group or idea about which data can be classified, collected and/or stored. Forms of entity used in this course are:

Database: an entity represents a table in a relational database that holds data.
System: the source or sink of the data which flows into or out of a system and over which the system has no control.

Error

Types of errors used in this course are:

- Syntax error: an error in the source code that does not meet the requirements of the specific programming grammar structure.
- Logical error: an error in the logic of an algorithm.
- Run-time error: an error occurring during the running of a program.

Executable code

Code which has been compiled into a low-level language program; for example, .exe, .com.

File transfer protocol (FTP)

Standard for transferring programs and data across a network.

Flow chart

Graphical representation of the sequence, selection and iteration flow within an algorithm.

Form

User interface for data entry, modification and query.

Function

User defined function is a sub routine designed for a specific task which receives data via parameters and returns a single value via the function name.

Gantt chart

A bar chart, emphasising time, used for scheduling projects.

Hypertext transfer protocol (HTTP)

Rules (protocols) governing the transfer of files (text, media, audio, video) across the Internet.

Identifier

User defined name of a program element, including variables, constants and arrays.

Integrity

Relates to the accuracy and consistency of the data. Primary areas, include: referential, entity and domain.

Keys

A key is a field, used in a database as a unique identifier. The types of keys used in this course are:

- Primary key: an attribute which uniquely identifies a record in a table.
- Composite key: a primary key consisting of two or more attributes.
- Foreign key: an attribute in a table which refers back to a primary key in a related table.

Malware

Malicious software designed to covertly access a system and cause harm.

Module

A block of code which can exist and run alone or can call other modules. Examples may include the main module, functions, or procedures.

Open source software

Collaboratively created software which is licensed to include modifiable source code.

Parameter
An argument which can be passed by value or by reference to a function or procedure or module.

Procedure
A sub routine designed to perform a specific task which does not return a value.

Project
Stimuli in the format of a case study or narrative presented to students undertaking a task, assignment or exam.

Project management
The management of a temporary task with defined start and end parameters that includes planning, budgeting, quality control, and/or human resources.

Protocol
Agreed formal descriptions of rules and formats used when communication/network devices exchange data.

Prototype
A model of a system produced using the iterative method involving design, create, and evaluate. Used in contrast to a formal SDLC method.

Pseudocode
Human readable description of the steps within a program, based on the algorithm.

Query
A method of interrogating a database to extract information. Examples include QBE and SQL.

Radio frequency identification (RFID)
Low cost self-powered RF tags designed to track items, such as animals on a farm or products in a shop or factory.

Redundant array of inexpensive devices (RAID)
Storage technology that divides and replicates data among multiple device drives.

Relation
A table within a (relational) database.

Report
The result of a query provided in a formalised format.

Simple mail transfer protocol (SMTP)
Internet standard protocol for transmitting (sending) email.

Software development cycle (SDC)
The formalised development structure imposed upon the creation of software.

Software licence
A legal instrument governing the Intellectual Property rights of the software creator.

Source code
The original readable code created by the programmer before compilation.

Standard operating environment (SOE)
The specification of hardware, operating systems and application software to be holistically applied across an office or organisation.

Statement
A line of source code.

String
A sequence of characters (often in quotation marks) normally consisting of alpha-numerics, symbols and/or spaces

Structured query language (SQL)
A (command line) database language that allows interrogation and manipulation of data using the following format:

- Select: specifies names of fields to be used in the query
- From: specifies the tables the data is contained in
- Where: specifies the criteria to be used to extract the data.

Syntax
The keywords and rules relating to a specific program language.

System
A set of elements or components that interact to accomplish a required outcome.

System boundary
An imaginary line separating the internal system from the outside elements.

Systems development life cycle (SDLC)
A linear system of defined stages, each of which requires completion before commencement of the following stage. The SDLC is costly, time consuming, highly documented and has little to no user input.

Topology
The physical or logical configuration of a network system.

Trace table
The manual testing of the logic of an algorithm.

Transmission control protocol/Internet protocol (TCP/IP)
TCP: a set of rules (protocols) used to transmit data packages across a network.
IP: a set of rules which allows the routing of data packages across a network.

Transmission media
The physical resources used to transmit data across a network, including cables or Wi-Fi.

Universal resource locator (URL)
The reference address to a web page (resource) on the internet.

Variable
Named memory location whose literal contents can change while the program is executed.