Stage 5 Information and Software Technology

Year: Year 9 **Teacher:**

Topic: Option 8: Software Development and Programming **Time:**

This option involves students undertaking a range of activities that will lead them to modifying and writing their own code when developing software products. Initially students will work with existing code to identify data types and control structures, leading to the development of algorithm descriptions.

Outcomes

A student:
5.2.1 describes and applies problem-solving processes when creating solutions
5.2.2 designs, produces and evaluates appropriate solutions to a range of challenging problems
5.2.3 critically analyses decision-making processes in a range of information and software solutions.

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Past, Current and Emerging Technologies				
• the impact of past, current and emerging information and software technologies on the individual and society including different cultural groups such as Aboriginal and Indigenous (core 2.1.0)	• explore and discuss current information and software technologies relevant to the option	Class discussion and comprehension exercise.	Wilson, C., 2001, *Exploring Computing Studies*, Second Edition, Cambridge University Press, pp. 71–72. Exercise 4.6.	
	• identify a variety of past, current and emerging information and software technologies	**Investigation:** Operating systems including Win95, 98, 2000, XP, Mac OS7-9, X, Linux, Unix and open source.	• Internet • Students	
	• evaluate the appropriateness of current and emerging information and software technology for specific purposes	**Activity:** Create a simple web page using *Dreamweaver* and *Microsoft Word* and compare the code of each of the applications. Students to evaluate.	• *Dreamweaver* • *Microsoft Word*	

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Basic programming concepts • input, process, output	• examine the code of an existing software program to describe the input, processes and output	**Minor project:** Examine a simple web page with graphics, forms and hyperlinks. Describe input/process and output of code.	• Internet	
• functions • assignment statements • variables • constants	• examine an existing program and identify functions, assignment statements, variables and constants • modify an existing program to assess the effects of changing variables on the output of the program	**Investigation:** Students view a simple HTML code and identify constants, variables and assignments.	• Internet • *Dreamweaver* • *Explorer*	
GUI layout including • graphics tools • objects such as textboxes, list boxes and command buttons • the function of the user interface (core 7.4.1.) • interactivity with the user (core 7.4.2)	• experiment with an existing GUI layout in a selected software program • design a simple GUI layout for a specific problem and apply simple programming code • conduct a peer evaluation on the designed GUI • explain the function of the user interface • compare and contrast types of user interfaces	**Discovery/problem-based learning minor project:** Students to use Internet to find good web site layouts that are icon-based. Students to carry out a PMI analysis of site. Students to design a simple web site that uses objects/icons.	Examples of icon-based web sites. • www.yahooligans.com • www.kartoo.com • www.seussville.com	

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Features and elements of a graphical user interface (GUI) such as (core 7.5)				
• functionality (core 7.5.4)	• explain the features and elements of GUI in a range of applications			
• instructions to the user (core 7.5.6)	• evaluate the effectiveness of GUI features and elements for a specific purpose			
Data types such as				
• character, integer, string, real, Boolean	• identify data types in existing code and explain their purpose • compare the use of data types	Classroom activity on data types and coding.	Wilson, C., 2001, *Exploring Computing Studies*, Second Edition, Cambridge University Press, pp. 93. Project exercise 1.	
Data coding such as				
• decimal and binary (core 3.3.1)	• describe and compare coding methods	**Activity:** Students to use decimal and binary converter to create a secret message to their friends in binary and decimal. Students try to figure out codes for alphabetical terms.	• Internet • http://nickciske.com/tools/binary.php	
• ASCII (core 3.3.2)	• perform simple calculations on data coding	Using HMTL demonstrate the use of ASCII coding for colour in HTML.	• Internet • http://www.free-cgi.com/freecgi/reference/ascii.asp	
Data operators				
• relational • logical • arithmetic operators	• distinguish between various operators within existing code	Classroom worksheet activity on operators.	Wilson, C., 2001, *Exploring Computing Studies*, Second Edition, Cambridge University Press, pp. 50–51. Exercise 3.6.	

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Algorithms				
• definitions and descriptions • representing algorithms	• define algorithms and describe examples in daily life	**Minor project:** Develop pseudocode or flow chart for sport choices.	• *Inspiration* • Worksheet • *Microsoft Word* (flowchart drawing) • Student knowledge and discussion.	
• examples such as recipes, directions, appliance instructions	• represent algorithms by using either flowchart or pseudocode • explain the purpose of an algorithm when solving problems	**Activity:** Create a list of instructions to create a paper aeroplane or to catch the train.		
People (Core) **Roles and responsibilities** of people working in the information and software technology field such as				
• systems analyst (core) • software engineers (core) • programmers (core)	• describe key roles within the information and software technology field and critically analyse possible role stereotypes • examine the contribution of people to the field of information and software technology • examine roles of people working in the field of information and software technology	Students to find newspaper advertisements for careers in programming, including system analyst, programmers and software engineers.	• Internet • www.9msn.com	
Careers in information and software technology (Core)				
• career paths (core)	• explore career opportunities and pathways for people within the field of information and software technology • discuss the use of information technology skills across industry and for self employment			

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Control structures • sequencing • selection such as binary and case • input/processes/output table (core 1.2.6) • repetition and/or iteration such as pre and post test	• devise algorithms to solve everyday problems incorporating the use of control structures • model possible solutions using a range of methods • examine and analyse the existing code of a selected example and identify control structures • develop prototypes using basic control structures	**Activity:** Turning on a computer sequence. **Selection:** use example of answering a phone or catching a train. Comprehension and discussion.	• Student knowledge. Wilson, C., 2001, *Exploring Computing Studies*, Second Edition, Cambridge University Press, pp. 71–72. Exercise 4.6. See pages 355–361 Lynch, I. J., 2000, *Concepts and Exercises in Computer Studies*, Exercise 2.	
Desk checking	• conduct a desk check on a selected algorithm • modify an algorithm to produce the required output	**Activity:** Pythagoras' Theorem that is incomplete. Students to test and make changes to solve.	Worksheet.	
Sub-programs • purpose	• examine existing code and algorithms to identify the purpose of sub-programs for a range of examples	**Micro worlds project:** Screen saver.	See *Micro worlds project*.	
• examples	• incorporate sub-programs into algorithms and working code			

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Programming language				
• function of programming language	• define and describe the function of a programming language	**Investigation/discovery learning:** Discussion of various types of programming languages including C, Fortran, HTML, Java, Pascal etc. Students to get examples of code.	• Internet	
• examples of a programming language	• convert algorithms into basic code using a given language syntax	• **Activity:** Identify the features of HTML. • What are tags? • Identification of common tags. • Research sites on the Internet that list HTML tags. • Investigate HTML: Create a document using an HTML editor Text Edit on Mac, Notepad on PC.	• Exercise 4 Questions 6, 7 and 8 on worksheets provided. • View the source code in their own web site in Dreamweaverˆ. • Use the interactive tutorial at APHS, Curriculum, Faculties, Vet, Computing and Technology, Year 9. • http://www.davesite.com/webstation/html/chap01.shtml	
Issues (core)				
• copyright and/or licensing (core 5.1.1) • piracy (core 5.1.2)	• examine legal issues as they apply to the development of information and software technology solutions	**Activity:** Find licence card for software product using Internet. Create a user-friendly guide to licence agreement in your own words.	• Web links • www.microsoft.com	

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Ethical issues such as				
• code of practice and conduct • privacy and security (core 5.2.1)	• research and report on ethical issues relating to the development of information and software technology solutions	Web Quest.	• Web Quest.	
• inappropriate use including hacking (core 5.2.3)	• identify the ethical responsibilities of software users	**Investigation:** Identify laws regarding computer hacking.	• Internet • http://www.police.nsw.gov.au/main/default.cfm • http://www.austlii.edu.au/forms/search1.html?&mask=au/legis/nsw/consol_act&	
Data structures				
• record	• examine data structures in existing code	Comprehension exercise.	Wilson, C., 2001, *Exploring Computing Studies*, Second Edition, Cambridge University Press, pp. 41–43. Exercise 3.1.	
• file	• demonstrate the use of an array	**Major project:** Micro worlds screen saver.	• See Major project: Micro worlds screen saver	
• array	• modify existing code to allow for changes to the array	**Major project:** Micro worlds screen saver.	• See Major project: Micro worlds screen saver	
Testing				
• test data	• test programming code using test data to check for the desired outcome	**Major project:** Micro worlds screen saver.	• See Major project: Micro worlds screen saver	

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Error detection including • syntax • logical • run-time	• identify and describe errors in a sample of given code • eliminate sources of error to create working code	**Major project:** Micro worlds screen saver	• See Major project: Micro worlds screen saver	
Error correction • software tools	• debug all errors in code using peer checking, desk checking or software debugging tools			
Documentation of programming code	• create appropriate user support documentation for code • apply meaningful variable names and comments to code			
Project development • processes and techniques • modifying an existing program • creating a new software solution • copyright and/or licensing (core 5.1.1) • piracy	• design, produce and evaluate a simple project for a real-world application either separately for this option, or integrated with other options • write code to solve a real-world problem • examine legal issues as they apply to the development of information and software technology solutions	**Major Project:** Screen saver. Students need to create an animated screensaver using Micro worlds. They are to create the source code using logo procedures including repetitions, variables, decision functions. Their animation should look as real as possible. Students should go through design, produce and evaluate cycle as well as provide evidence of testing in a log. It should require some user input variables.	• Worksheet. • *Micro worlds* software. • Logo tutorial.	

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
• (project) functionality of solution (Core)	• establish criteria for the evaluation of solutions			
• written (core1.7.2)	• outline a range of communication techniques appropriate to the solution			
• graphical and visual (core 1.7.3)	• communicate ideas, processes and solutions to a targeted audience			
• producing the solution (core 1.3.1)	• develop and implement the stages involved in the completion of a solution • apply set criteria to choose the most appropriate solution			

Additional content

Students learn about:	Students learn to:	Teaching and learning strategies	Resources	Registration
Random and sequential files **Object-oriented software development**	• compare and contrast existing code for processing random access and sequential files • investigate developments of object-oriented software development	**Research:** Students to investigate software companies that specialise in the development of software specific to client needs.	• Internet • www.download.com *Microsoft Word*	
Software (core) **Types and examples of software** • application including – customised	• compare and contrast the features of packages, including relationships to other packages • list the features of software packages appropriate to particular users and a range of tasks	**Activity:** Student is to download trial software for purpose of comparison. **Activity:** Student lists software features and makes a recommendation to a fictional client.		

Life skills

For some students with special education needs, particularly those students with an intellectual disability, it may be determined that the above content is not appropriate. For these students, Life Skills outcomes and content can provide the basis for the development of a relevant and meaningful program. See section 8.