

Curriculum notes Years 9-13:

A streamed comparison of prescriptions for ICT skill-based coursework

Vince Brannagan, Wellington High School

[From: Case study BP624 Interactive programming project for school-based clients – <http://www.techlink.org.nz/GIF-tech-education/beacon-practice/ICT/>]

Skills included in Information Science courses at Wellington High School are selected in line with the following publications, which include a diverse range of skills that can be taught to students with a wide range of abilities and personal goals.

- A Model Curriculum for K–12 Computer Science – Final Report of the ACM K–12 Task Force Curriculum Committee October 2003, Association for Computing Machinery, Inc, United States.
- University of Cambridge International Examinations:(CIE) Computer Studies 7010 GCE O Level and Computing 9691 GCE A/AS Level England
- International Computer Driving Licence (ICDL) Syllabus Version 4.0
- A proposal for an International Olympiad in Informatics Syllabus, Tom Verhoeff, Gyula Horv'ath, Krzysztof Diks and Gordon Cormack.
- NZIST Draft Curriculum Build 004 October 1 2001.
- Waikato University Computer Science Scholarship Examination Syllabus.

Year	K-12 (Copyright ©2006 by the Association for Computing Machinery, Inc.	ICDL
9	Level I—Foundations of Computer Science <i>Grades 6–8:</i> Upon completion of grade 8, students will: 1. Apply strategies for identifying and solving routine hardware and software problems that occur during everyday use. 2. Demonstrate knowledge of current changes in information technologies and the effects those changes have on the workplace and society. 3. Exhibit legal and ethical behaviors when using information and technology and discuss consequences of misuse. 4. Use content-specific tools, software, and simulations (e.g., environmental probes, graphing calculators, exploratory environments, Web tools) to support learning and research. 5. Apply productivity/multimedia tools and peripherals to support personal productivity, group collaboration, and learning throughout the curriculum. 6. Design, develop, publish, and present products (e.g., Web pages, videotapes) using technology resources that demonstrate and communicate curriculum concepts to audiences inside and outside the classroom. 7. Collaborate with peers, experts, and others using telecommunications tools to investigate educational problems, issues, and information, and to develop solutions for audiences inside and outside the classroom. 8. Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems. 9. Demonstrate an understanding of concepts underlying hardware, software, algorithms, and their practical applications. 10. Discover and evaluate the accuracy, relevance, appropriateness, comprehensiveness, and bias of electronic information sources concerning real-world problems. 11. Understand the graph as a tool for representing problem states and solutions to complex problems. 12. Understand the fundamental ideas of logic and its usefulness for solving real-world problems.	See year 11
10		

Year	k-12 Copyright ©2006 by the Association for Computing Machinery, Inc.	CIE O Level	ICDL	NZIST Draft Curricula
11	<p>Level II—Computer Science in the Modern World</p> <p>1. Principles of computer organization and the major components (input, output, memory, storage, processing, software, operating system, etc.)</p> <p>2. The basic steps in algorithmic problem solving (problem statement and exploration, examination of sample instances, design, program coding, testing and verification)</p> <p>3. The basic components of computer networks (servers, file protection, routing protocols for connection/communication, spoolers and queues, shared resources, and fault-tolerance).</p> <p>4. Organization of Internet elements, Web page design (forms, text, graphics, client- and server-side scripts), and hypermedia (links, navigation, search engines and strategies, interpretation, and evaluation).</p> <p>5. The notion of hierarchy and abstraction in computing, including high-level languages, translation (compilers, interpreters, linking), machine languages, instruction sets, and logic circuits.</p> <p>6. The connection between elements of mathematics and computer science, including binary numbers, logic, sets, and functions.</p> <p>7. The notion of computers as models of intelligent behavior (as found in robot motion, speech and language understanding, and computer vision), and what distinguishes humans from machines.</p> <p>8. Examples (like programming a telephone</p>	<p>Section 1</p> <p>Applications of computers and their social and economic implications The aim of this section of the syllabus is to cover as broad a range of computer applications as possible, so as to provide an understanding of the power and versatility of the computer and the benefits of its use, and also of its limitations and the problems and potential disadvantages of its use.</p> <p>1.1 The range and scope of computer applications Candidates should be able to demonstrate knowledge and understanding of a wide variety of computer applications. They will not be expected to have specific knowledge of every kind of application, but should be able to comment sensibly on any suggested application and make use of specific relevant examples for illustration. General application areas Communication and information systems, on-line services and remote databases Commercial and general data processing Industrial, technical and scientific uses Monitoring and control systems Automation and robotics Expert systems and artificial intelligence The range of computer applications is vast. An awareness of this range and of the nature of a variety of application areas is best achieved through a general study of a number of applications of different kinds. The relevant aspects of any application include the following:</p> <ol style="list-style-type: none"> 1 The purpose of the application 2 The required outcome 3 The overall system design, including both the computerised and the non-computerised parts of the application 4 The necessary inputs to the system and the means by which any data is captured 5 The overall organisation and processing of the data within the system 6 The use and organisation of the major software and hardware components of the system 7 The need for recovery in the event of a system failure 8 The interface between the system and its users 9 The effectiveness of the system in practice 10 The effects of the application on individuals and organizations. <p>There are many examples in each application area and some are listed below. Candidates will be expected to have studied a varied sample of these. Examples could be selected from electronic mail, fax, electronic/video conferencing, digital telephone facilities, information retrieval and database systems, office automation and library systems, viewdata systems, multimedia systems, e-commerce, wireless technology, the internet and virtual reality. Examples could be selected from banking systems, hospital administration, systems for personnel records, stock control and order processing...</p> <p>1.2 The social and economic implications of the use of computers In addition to knowledge about applications, candidates should be able to demonstrate a broad knowledge of the economic reasons for using computers and the effects of their use across a range of application areas. They should be able to formulate a reasoned view of the potential effects of any suggested application, or development, and to show their critical abilities in balancing the advantages and disadvantages of a computerised system. Social and economic effects on people and organisations associated directly with the application, on other individuals and organisations, and on society in general Economic reasons for the use of computers Changes to existing methods, products and services Development of new products and services in the working environment Changes in employment, retraining Privacy and integrity of data Data protection legislation Security and reliability Consequences of system failure Hacking and other computer crime Computer viruses</p> <p>The social and economic effects of the use of computers should be discussed in the context of particular computer applications, with any general principles being based upon real examples. For example, the de-skilling brought about through the replacement of skilled and semi-skilled labour by micro-processor-controlled systems in manufacturing; the process of electronic scabbing which allows managers to switch word-processing duties from striking clerks in one country to non-strikers in another...</p> <p>Candidates may see the need for constant retraining of staff as existing packages are upgraded and new ones</p>	<p>Module 1 Concepts of Information Technology (IT), requires the candidate to have an understanding of some of the main concepts of IT at a general level. The candidate is required to understand the make-up of a personal computer in terms of hardware and software and to understand some of the concepts of Information Technology (IT) such as data storage and memory. The candidate shall also understand how information networks are used within computing and be aware of the uses of computer-based software applications in everyday life. The candidate shall appreciate health and safety issues as well as some environmental factors involved in using computers. The candidate shall be aware of some of the important security and legal issues associated with using computers.</p> <p>Module 2 Using the Computer and Managing Files, requires the candidate to demonstrate knowledge and competence in using the common functions of a personal computer and its operating system. The candidate shall be able to adjust main settings, use the built-in help features and deal with a non-responding application. He or she shall be able to operate effectively within the desktop environment and work with desktop icons and windows. The candidate shall be able to manage and organize files and directories/folders, and know how to duplicate, move and delete files and directories/folders, and compress and extract files.</p>	<p>Computer Systems Components of a computer; CPU, RAM, ROM, peripherals. Memory; basic features: quantity. Printers; line vs page. Software: Applications operating system; File management; drive paths, folders, file name copy, move, rename, delete.</p> <p>Computer Application Word Processing – Create, edit, save and print simple one-column documents of one or two pages. These will contain formatted text, and require settings for margins, paper size and orientation. Numbered and/or bulleted lists will be included. Simple tables will be included. Headers and footers Drawing to Spreadsheets – single worksheet, simple formulas, simple functions (SUM), simple formatting (\$,%), one-dimensional replication, simple graphics according to given instructions. Databases - simple flat file database, specified design, simple query, simple columnar report Desktop Publishing - Simple one-page, one-column documents with one or more graphics. Text flow around graphics and between columns or frames. Graphics Software – Single drawn and paint images. May be inserted into other documents. Web Page Design – a small set(2-3) of single</p>

Year	k-12 Copyright ©2006 by the Association for Computing Machinery, Inc.	CIE O Level	ICDL	NZIST Draft Curriculu
	<p>answering system) that identify the broad interdisciplinary utility of computers and algorithmic problem solving in the modern world.</p> <p>9. Ethical issues that relate to computers and networks (including security, privacy, intellectual property, the benefits and drawbacks of public domain software, and the reliability of information on the Internet), and the positive and negative impact of technology on human culture.</p> <p>10. Identification of different careers in computing and their connection with the subjects studied in this course (e.g., information technology specialist, Web page designer, systems analyst, programmer, CIO).</p>	<p>published. Candidates may appreciate the use of individualised training packages that use CD-ROM and multimedia.</p> <p>The requirements for security and reliability vary considerably depending on the nature of the application. For example, a failure during a batch update of a sequential master file is irritating and will cause delay, whereas a failure in an air traffic control system could well have catastrophic results.</p> <p>Computer crime includes activities such as the cracking of ineffective security systems so as to gain unauthorized access to commercially sensitive or confidential personal files, and fraud through the improper transfer of funds from one account to another. Computer criminals may work within the organisation suffering the crime or may be outsiders. Measures taken to combat computer crime include physical security, development of complex security codes and systems, encryption of sensitive data, and monitoring of all attempts to access the system, whether successful or not.</p> <p>Sensible precautions should be identified.</p> <p>Section 2 Analysis of the system</p> <p>The aim of this section of the syllabus is to cover the main principles of the analysis of the system which are problem definition, feasibility study, investigation and fact finding. This is achieved partly through the study of computer applications, in particular the methods by which a problem has been analysed to lead to a successful solution for the user, and partly through practical work.</p> <p>Systems analysis</p> <p>Candidates should be able to describe the main steps involved in systems analysis and they should be able to use charts and diagrams. They should be able to apply the principles of systems analysis in their Course Work projects. Identification of the problem and stating it briefly. Deciding and stating specific outcomes which are desired in the solution of a particular problem. Analysing the flow of information and data in existing (computer and manual) solutions. Evaluation of existing solutions. Consideration of alternative solutions. A variety of problems/solutions may be examined beginning with the relatively simple and proceeding to the more complex. The aim should be to develop a feeling for standard stages which would be appropriate during systems analysis. The ability to describe a problem and its solution in nontechnical terms should be developed in a similar way.</p> <p>Section 3 Problem solution including algorithm design and programming concepts. The aim of this section of the syllabus is to cover the design, development, implementation, maintenance and review principles, which include techniques and tools which relate to the solution to a problem. A study of these topics is reinforced through practical work and illustrated by a consideration of existing problem solutions in computer applications.</p> <p>Making an overall plan.</p> <p>Most of this work is encountered through practical exercises. Defining the scope of separate modules Designing algorithms which relate clearly to the requirements of the system Explaining algorithms and how they relate to the system Explaining how hardware needs arise from the output required from the system</p> <p>Algorithm tools Interpreting and testing algorithms</p> <p>Top-down design, structure diagrams, flowcharts, menus, libraries of procedures and subroutines. Candidates should be able to work out the purpose of an algorithm, perhaps with the help of a dry run, and to suggest and apply suitable test data.</p> <p>3.2 Candidates should have experience of representing algorithms informally (as structure diagrams, flowcharts, step sequences, structured English) and more formally in pseudocode. Candidates should appreciate the main requirements of a programming language, to allow manipulation of data of various types and structures, including control of input and output, and to provide for selection, repetition and subprogram intercommunication. Candidates should have a simple understanding of the functions of interpreters, compilers and assemblers, and an appreciation of the benefits offered by the existence of a range of languages, both high and low level. The concept of a program</p> <p>High-level languages Low-level languages Pseudocode structures Documentation User documentation and technical documentation</p> <p>Candidates will not be expected to code in any particular</p>	<p>The candidate shall also understand what a computer virus is and be able to use virus-scanning software. The candidate shall demonstrate the ability to use simple editing tools and print management facilities available within the operating system.</p> <p>Module 3 Word Processing, requires the candidate to demonstrate the ability to use a word processing application on a computer. The candidate shall be able to accomplish everyday tasks associated with creating, formatting and finishing small sized word processing documents ready for distribution. He or she shall also be able to duplicate and move text within and between documents. The candidate shall demonstrate competence in using some of the features associated with word processing applications such as creating standard tables, using pictures and images within a document, and using mail merge tools.</p> <p>Module 4 Spreadsheets, requires the candidate to understand the concept of spreadsheets and to demonstrate the ability to use a spreadsheet application on a computer. The candidate shall understand and be able to accomplish tasks associated with developing, formatting, modifying and using a spreadsheet of limited scope ready for distribution. He or she shall also be able to generate and apply standard mathematical and logical formulas using standard formulas and functions. The candidate shall demonstrate competence in creating and formatting graphs/charts.</p> <p>Module 5 Database, requires the candidate to understand some of the main concepts of</p>	<p>pages linked together.</p> <p>Computer Communications</p> <p>The Internet – basic concepts Web pages; creating a using; hyperlinks Using the Internet – UF Using email; attachmer Precautions when using the Internet; viruses, security (persc and data). Computers and People Employment opportunit Recognition of the effect of computers on societ Privacy and Copyright; concepts Simple case studies Simple ergonomics Computer languages; examples e types; Computer Programm Simple problem analys and documentation; simple flow charts and structure diagrams Concept of an ordered of instructions Concept of a “comman constructed from primitives (procedure) Concept of a function v procedure Creating procedures (c functions) to address specific parts of a program’s functionality Concept of data types Concept of variables to store data; concept of scope Concept of repetition b conditional (Repeat...Until, Do..While) and unconditional (For...Ne Documenting a subrou</p>

Year	k-12 Copyright ©2006 by the Association for Computing Machinery, Inc.	CIE O Level	ICDL	NZIST Draft Curriculu
		<p>language but they should be familiar with the concepts of sequence, selection and repetition. Repeat ... Until. If ... Then ... Else ... Endif. Case of Otherwise ... Endcase. While ... Do ... Endwhile.</p> <p>Candidates should be able to distinguish between the documentation required by users and that required by those responsible for improving and maintaining a solution in working order or for developing the solution to meet new needs.</p> <p>Section 4 Generic software and the organisation of data The aim of this section is to acquaint the candidates with a broad view of generic software packages. It is expected that these will be experienced through practical work. Although candidates may choose to become expert in the use of a particular package, only a general knowledge is required of the kinds of features typical of generic packages.</p> <p>4.1 Software: wordprocessing, database management, spreadsheets, graphics, communications, multimedia, data logging, computer aided design, programming desktop publishing and web design How applications packages solve sets of standard problems. Typical problems capable of solution by packages. Use of standard techniques or routines for established forms of processing (for example, file processing, sorting, merging simulation) Candidates should appreciate ways including import and export of using programming, desktop publishing, datalogging, word-processing packages, spreadsheets, databases, graphics packages, information retrieval packages, and show an understanding of the use of graphical user interfaces, communications software (including email), web browsers and search engines and authoring packages. They should appreciate the virtues and disadvantages of integrated packages, and of generic packages compared with specially written software and how packages may be customised by the use of macros. They should be able to explain the use of the tools associated with data logging.</p> <p>4.2 Data. Candidates should be aware of the standard methods of data collection, verification and validation, and where it is appropriate to use particular methods. Candidates should understand that data, particularly data held in files, requires access in different ways depending on the particular application. The medium on which the data is stored, and particularly the way in which it is organised, depend on the requirements for access. Candidates should be familiar with the idea of file ordering and the principles of sorting and merging. They should have experience of sequential file processing and of processing individual records by means of record keys. The relationship between information and data; the collection of data, methods of ensuring its correctness (including validation and verification and the distinction between these); the coding of data for input; the presentation of useful information from processed data; analogue-to-digital and digital-to-analogue conversions File organisation: different forms of organisation, depending on the data stored and the requirements for processing; processing methods Data types: numbers, characters, strings arrays, the need for different data types and structures to represent the data of problems which are being solved using a computer Candidates should be able to describe methods of data collection, verification, validation and presentation, and give examples of analogue-to-digital and digital-to-analogue conversion. Candidates should be able to select and justify appropriate methods of data collection, verification, validation and presentation, and assess the use of physical variables such as temperature and pressure to control processing activities. Candidates should be able to describe the need for and the simple processing of files, and select, with reasons, appropriate file organisation and processing methods for a particular application, and understand routines used for file maintenance such as updates, additions and deletions. Candidates should be able to identify the different forms of data and representations for processing which relate to a given simple problem, and explain the need for different data types and structures and how these relate to the data of a given problem.</p> <p>Section 5 Hardware, systems and communications The aim of this section is to draw together the experience of various kinds of hardware and types of processing, and to examine the concepts of operating systems and communications.</p> <p>5.1 Hardware Computer, microcomputer, microprocessor, standard input and output devices, broad classes of processor power The functions and characteristics of storage media The characteristics and performance of a range of peripheral equipment (including control devices) Candidates should be able to identify the use of microprocessors in everyday objects such as cameras, digital watches, etc. Candidates should be able to describe suitable input and output</p>	<p>databases and demonstrate the ability to use a database on a computer. The candidate shall be able to create and modify tables, queries, forms and reports, and prepare outputs ready for distribution. The candidate shall be able to relate tables and to retrieve and manipulate information from a database by using query and sort tools available in the package.</p> <p>Module 6 Presentation, requires the candidate to demonstrate competence in using presentation tools on a computer. The candidate shall be able to accomplish tasks such as creating, formatting, modifying and preparing presentations using different slide layouts for display and printed distribution. He or she shall also be able to duplicate and move text, pictures, images and charts within the presentation and between presentations. The candidate shall demonstrate the ability to accomplish common operations with images, charts and drawn objects and to use various slide show effects.</p> <p>Module 7 Information and Communication, is divided in two sections. The first section, Information, requires the candidate to understand some of the concepts and terms associated with using the Internet, and to appreciate some of the security considerations. The candidate shall also be able to accomplish common Web search tasks using a Web browsing application and available search engine tools. He or she shall be able to bookmark Web sites, and to print Web pages and search outputs. The candidate shall be able to navigate within and complete Web-based forms. In</p>	

Year	k-12 Copyright ©2006 by the Association for Computing Machinery, Inc.	CIE O Level	ICDL	NZIST Draft Curriculu
		<p>devices in relation to the requirements of applications. The functions and characteristics of storage media such as RAM, ROM, CD-ROM, DVD, discs and tapes need to be considered in relation to the requirements of applications. Candidates should be able to discuss the suitability of different peripherals for various applications.</p> <p>5.2 Systems and communications Operating system facilities: The nature of batch, online, multi-access, real-time transaction processing, multitasking, network and process control operating systems; The form of interface between the operating system and the user; command line and use of graphical user interfaces; Management of files; file directories; Peripheral device control; use of buffers; interrupts and interrupt priorities; polling; handshaking; check sums. A real-time transaction processing system is an online system in which individual, discrete transactions are processed as they occur; an airline booking system and an on-line stock control system are typical examples. This use of the term real time differs from its use in a real-time process-control system, in which physical quantities are continuously monitored and processed sufficiently rapidly to be capable of influencing the sources of data. The need for and use of facilities to copy, move, list, print files. The use of sub-directories. For an interrupt system the external device or event interrupts the processor, whereas in a polling system the processor interrogates the device or status register. These two alternative methods have very different processing requirements. Adding together all of the elements (for example, bytes) of a block produces a single element known as the check sum. This can then be stored with the block and provides a check when the block is transferred.</p> <p>5.3 Types of system Candidates should be able to distinguish between the different types of system, to describe what is needed to support them, to explain which is the most suitable for any given application and to discuss the consequent implications for the user. Candidates should be able to describe particular problems in the management of the various types of system, such as conflicting access to common data or critical timing considerations. Batch processing systems Single-user online systems Multi-user online systems Network systems Control systems Automated systems Multimedia In broad terms, a batch processing system is one in which a job is processed without any direct interaction between the job and the user. Typical applications are payroll and billing systems. In contrast to this, an online system provides for interaction between the job and the user, which may influence the future course of processing. Such systems may be single-user (for example, a personal computer) or multi-user. Typical applications are word processing and online information retrieval. A network system is one in which processing is carried out independently in more than one location, but with shared and controlled access to some common facilities which normally include file storage and information resources. Star networks, linear networks and ring networks should be compared. Candidates should understand the need for local and shared resources and gateways, for accessing the wide area network (WAN). In a control system, one or more computers is used to control the operation of some non-computer equipment, usually involving some monitoring and logging of physical quantities, providing some analysis of performance and allowing some user interaction. Feedback is an essential element in most control systems. Timing considerations are often critical and the term real-time control system is sometimes used to indicate this. Control systems are used in applications such as oil refining, chemical processing and integrated traffic-control systems. Automated systems are broadly similar to control systems but are dedicated to a particular task, and lack the ability to collect and analyse data and the flexibility to allow for and act on user interaction beyond a very simple level. Examples are the systems found in equipment such as washing machines and cameras. Candidates should be able to specify minimum hardware and software requirements for multimedia applications and describe typical features and uses of multimedia systems.</p>	<p>the second section, Communication, the candidate is required to understand some of the concepts of electronic mail (e-mail), together with having an appreciation of some of the security considerations associated with using e-mail. The candidate shall also demonstrate the ability to use e-mail software to send and receive messages, and to attach files to mail messages. The candidate shall be able to organize and manage message folders/directories within e-mail software.</p>	

Year	k-12 Copyright ©2006 by the Association for Computing Machinery, Inc.	CIE A/AS level	ICDL	NZIST Draft Curriculum
12	<p>Level III—Computer Science as Analysis and Design</p> <p>A. Fundamental ideas about the process of program design and problem solving, including style, abstraction, and initial discussions of correctness and efficiency as part of the software design process.</p> <p>2. Simple data structures and their uses</p> <p>3. Topics in discrete mathematics: logic, functions, sets, and their relation to computer science</p> <p>4. Design for usability: Web page design, interactive games, documentation</p> <p>5. Fundamentals of hardware design</p> <p>6. Levels of language, software, and translation: characteristics of compilers, operating systems, and networks</p> <p>7. The limits of computing: what is a computationally “hard” problem? (e.g., ocean odelling, air traffic control, gene mapping) and what kinds of problems are computationally unsolvable (e.g., the halting problem)</p> <p>8. Principles of software engineering: software projects, teams, the software life cycle</p> <p>9. Social issues: software as intellectual property, professional practice</p> <p>10. Careers in computing: computer scientist, computer engineer, software engineer, information technologist</p>	See year 13	See year 11	<p>Computer Systems</p> <p>Concepts of computer speed and bandwidth; caching, Technology used in peripheral devices – printers, CDROM, LCD screen</p> <p>Concepts of primary and secondary memory</p> <p>Concept of a print buffer and a print queue</p> <p>Common printer technologies</p> <p>Choosing appropriate equipment;</p> <p>Basic networking; topologies, speeds, components, software;</p> <p>Mapping and sharing network resources; folders, drives and printers;</p> <p>Functions of an operating system vs functions of an application</p> <p>Network security; permissions</p> <p>Understanding the limitations of some Oses at a simple level.</p> <p>Computer Applications</p> <p>Word Processing – multi-page, multi-columnar, multi-section documents. Page layout will vary within the document. Embedded graphics and the use of tabs are expected.</p> <p>Mail-merge.</p> <p>Spreadsheets – two-dimensional replication, more complex formatting (cell merge, alignment, date/time, fixed decimal, type styles, cell borders, use of colour), select appropriate chart or graph style, conditional functions, named ranges.</p> <p>Databases – Flat-file, Field selection and design, data entry forms, query design and use, Boolean queries, grouped reports, summary variables, expressions and functions</p> <p>Relational databases</p> <p>Desktop Publishing – multiple pages, master pages, layered objects, drawn objects, headers and footers.</p> <p>Graphics Software – Complex images created from combining together simpler image using appropriate effects; specific issue or problem in mind.</p> <p>Web Page Design - a small site designed as a unit, with overall styles and design parameters.</p> <p>Computer Communications</p> <p>The Internet; structure; clients, servers, protocols, IP addresses</p> <p>Email; clients, servers, protocols; (concept of a protocol)</p> <p>Web sites; design and creation; simple HTML tags</p> <p>Presentations for imparting information.</p> <p>Computers and People</p> <p>People who work with computers; what do they do?</p> <p>Detailed analytical case study of a worker/workplace</p> <p>Impacts of computer use; trends; future predictive analysis</p> <p>Legislation concerning computers</p> <p>Ethical behaviour; situational studies</p> <p>Software types, rights and responsibilities</p> <p>Observational case studies of workplaces and the roles of workers in them</p> <p>Considering people as part of the computer system</p> <p>Further ergonomics; OOS</p> <p>Computer Programming</p> <p>Character codes; binary and hexadecimal numbers</p> <p>Concepts and examples of logic gates, adders, etc.</p> <p>Creating a complete program to perform a simple task</p> <p>Concept of libraries, reusable code and eventually objects</p> <p>Concept of data structures (arrays, records, lists, stacks, heaps). Concept of event-driven interfaces. Reading and writing line-structured text files. Manipulating data read from a file. Documenting a program. Working from a supplied brief Creating and worki from a plan of action</p>

Year	k-12	CIE A/AS level	NZIST Draft Curriculum	Waiakato	IOI
13	<p>Level IV—Topics in Computer Science</p> <p>These electives include, but are not necessarily limited to:</p> <ul style="list-style-type: none"> • Advanced Placement (AP) Computer Science • A projects-based course in which students cover a topic in depth. • A vendor-supplied course, which may be related to professional certification. 	<p>Section 1: Computer Systems, Communications and Software is the foundation for all subsequent sections. It provides candidates with an understanding of the core aspects of computer systems, which is developed and enhanced in subsequent sections.</p> <p>Section 2: Practical Programming Project requires candidates to demonstrate their skills in a programming language by selecting a problem to solve. The solution to the problem should encompass as many of the criteria listed in Section 2 of the syllabus as the candidate is capable of using. Candidates and Centres should be aware that demonstration of the skills will be necessary, within the context of the problem solution, in order to earn marks in the assessment. It is envisaged that this project will be a long term piece of work to be completed during year one of the course. It will be submitted for moderation in the same session that the candidate offers Paper 1. In this way,</p>	<p>Computer Systems Concepts associated with processor design (RISC vs CISC), bus speed, multiprocessing, multitasking, parallel processing. Concepts associated with network design and operation; simple network maintenance and operation tasks. Batching network tasks. Selecting and supplying hardware; what, why and where from. Encryption; why used; how it works (PKI); Computer Applications Word Processing – Full multi-page document, with embedded material from other applications, complete with table of contents is expected. Spreadsheet – multi-page workbooks, multi-page cell references, expression builder, multi-variable charts or graphs, interpretation of charts or graphs, frequency tables, logical and financial functions. Databases – Flat-file, automated data entry forms, macros, design and help notes, multiple reports, problem-solving focus. Custom user interfaces and command buttons Desktop Publishing - multiple document types with a problem-solving focus. Could also include (optional): Project Management - Use of project management software to manage a small project. Full documentation required. Graphics Software – Complex multiple images</p>	<p>Programming topics</p> <p>Control structures: Sequence Iteration Choice Nested Structures Control statements: If-then-else While Do-while(or repeat-until) Switch(or case) Data Types: Integer Real(or float) Character Boolean Data structures: Arrays(one and two dimensional) Strings(or character arrays) Data operations: Variable assignment Arithmetic expressions Boolean expressions Operator expressions Input/output: Input operations Output Formatting Reading and writing text files Printable and non-printable characters Programming structure: Procedures Functions Value parameters Built-in functions(or library routines) Algorithms and problem-solving: Examples.</p> <p>Computing concepts</p> <p>Base conversion: Conversion of positive integers between bases binary, octal, decimal. Binary numbers and arithmetic: Addition and multiplication of numbers in unsigned format. Subtraction</p>	<p>Mathematics</p> <p>Numbers and Geometry</p> <p>Integers, operations (incl. exponentiation), comparison Properties of integers (positive, negative, even, odd, divisible, prime) Fractions, percentages Point, vector, Cartesian coordinates (on a 2D integer grid) Euclidean distance, Pythagoras' Theorem Line segment, intersection properties Angle Triangle, rectangle, square Polygon (vertex, side/edge, simple, convex, inside/outside)</p> <p>Discrete Structures</p> <p>Functions, relations, and sets: Functions (surjections, injections, inverses, composition) Relations (reflexivity, symmetry, transitivity, equivalence relations, total/linear order relations, lexicographic order) Sets (Venn diagrams, complements, Cartesian products, power sets) Pigeonhole principle Excluded: Cardinality and countability (of infinite sets) Basic logic: Propositional logic Logical connectives (incl. their basic properties) Truth tables Predicate logic Universal and existential quantification Modus ponens and modus tollens Proof techniques: Notions of implication, converse, inverse, contrapositive, negation, and contradiction Direct proofs, proofs by: counterexample, contraposition, contradiction Mathematical induction Strong induction (also known as complete induction) Recursive mathematical definitions (incl. mutually recursive definitions) Basics of counting: Counting arguments (sums and product rule, inclusion-exclusion principle, arithmetic and geometric progressions, Fibonacci numbers) Pigeonhole principle (to obtain bounds) Permutations and combinations (basic definitions) Factorial function, binomial coefficient Graphs and trees: Trees (connected, no cycles, #nodes = #edges + 1; ordered/not-ordered) Undirected graphs (degree, path, cycle, connectedness, Euler/Hamilton path/cycle, handshaking lemma) Directed graphs (in-degree, out-degree, directed path/cycle, Euler/Hamilton path/cycle) Spanning trees Traversal strategies (defining the node order for ordered trees) 'Decorated' graphs with edge/node labels, weights, colours Multigraphs, graphs with self-loops</p> <p>Computing Science</p> <p>Programming Fundamentals</p> <p>Fundamental programming constructs: (for abstract machines) Basic syntax and semantics of a higher-level language (also see the following Variables, types, expressions, and assignment Simple I/O</p>

		<p>Centres that choose to do Papers 1 and 3 at the end of year 2 of a two year course, can do one piece of coursework in each year. It is envisaged that most candidates will use one of the languages listed on page 3. Other languages are welcomed, though CIE Customer Services should be consulted before the project is started.</p> <p>Section 3: Systems Software Mechanisms, Machine Architecture, Database Theory, Programming Paradigms and Integrated Information Systems provides candidates with further ability to extend skills, knowledge and understanding of computing concepts, gained in Section 1, to a range of applications in which computer systems are used.</p> <p>Section 4: Computing Project requires candidates to identify a well-defined user-driven problem, involving a third-party user, and to generate a solution. As for Section 2, this is done using software tools chosen by the candidate and may include a programming language, an appropriate applications package or other</p>	<p>to address an issue or solve a problem. Blending or morphing (or other appropriate techniques) to be used; animation may be used instead of or as well as other techniques. File formats and compression</p> <p>Computer Communications Networking; protocols (TCP/IP and HTTP – concepts), hardware, topologies and cabling HTML, XML, CSS, scripting</p> <p>Presentations for problem-solving; embedded applications Multimedia applications Computers and People</p> <p>Creating help files or documentation to assist a specific individual or group with a specific task</p> <p>Computer Programming</p> <p>This section is much more student and problem-centred, and so the emphasis is on the application of learning to date rather than on the demonstration or acquisition of new skills. Programming to solve a problem; writing code to a specific purpose and with specific constraints</p> <p>Documenting both program and process; Completion of a brief and a plan of action</p>	<p>using two's complement representation. Type representation: Students should know how characters might be represented using (for example) ASCII, UNICODE..</p> <p>Representation of signed integers and fixed point numbers in two's complement. Floating point numbers using two's complement notation only (b-bit fraction two's complement for mantissa, signed integer for exponent). Awareness of errors associated with accuracy and loss of precision- underflow, truncation and overflow.</p> <p>Normalisation of floating point numbers is not required.</p> <p>Language: Compilers and interpreters for high level languages. Computer architecture: Features of commonly used input, output and backing store devices. Structure of a microprocessor (ALU, CU- internal details and registers not required). Function of RAM, ROM, cache memory.</p> <p>The following terms should be understood: Virtual memory Primary and secondary memory Bit(b), byte(B) frequency (Hz) and their modifiers (k, M,G,T)</p> <p>Serial and direct access</p> <p>Students should be able to outline recent developments in processor architecture,</p>	<p>Conditional and iterative control structures</p> <p>Functions and parameter passing</p> <p>Structured decomposition</p> <p>Algorithms and problem-solving: Problem-solving strategies (understand–plan–do–check, separation of concerns, generalization, specialization, case distinction, working backwards; The role of algorithms in the problem-solving process</p> <p>Implementation strategies for algorithms (also see §6.3 SE1)</p> <p>Debugging</p> <p>The concept and properties of algorithms (correctness, efficiency)</p> <p>Fundamental data structures: Primitive types (Boolean, integer, character) Arrays (incl. multidimensional arrays) Records</p> <p>Strings and string processing</p> <p>Static and stack allocation (elementary automatic memory management)</p> <p>Linked structures (linear and branching)</p> <p>Static memory implementation strategies for linked structures</p> <p>Implementation strategies for stacks and queues</p> <p>Implementation strategies for graphs and trees</p> <p>Strategies for choosing the right data structure</p> <p>Abstract data types, priority queue, dynamic set, dynamic map</p> <p>Recursion: The concept of recursion Recursive mathematical functions Simple recursive procedures (incl. mutual recursion)</p> <p>Divide-and-conquer strategies</p> <p>Recursive backtracking</p> <p>Algorithms and Complexity</p> <p>Algorithms are fundamental to computer science and software engineering.</p> <p>Basic algorithmic analysis: Algorithm specification, precondition, postcondition, correctness, invariant</p> <p>Asymptotic analysis of upper complexity bounds (i.e. worst-case; informally)</p> <p>Big O notation</p> <p>Standard complexity classes (constant, logarithmic, linear, $O(N \log N)$, quadratic, cubic, exponential)</p> <p>Time and space tradeoffs in algorithms</p> <p>Algorithmic strategies: Simple loop design strategies</p> <p>Brute-force algorithms (exhaustive search)</p> <p>Greedy algorithms (insofar that understanding correctness is elementary)</p> <p>Divide-and-conquer (insofar that understanding efficiency is elementary)</p> <p>Backtracking (recursive and non-recursive)</p> <p>Branch-and-bound (insofar that understanding correctness and efficiency are elementary)</p> <p>Pattern matching and string/text algorithms (insofar that understanding correctness and efficiency are elementary)</p> <p>Dynamic programming</p> <p>Fundamental computing algorithms: Simple numerical algorithms involving integers (Euclid's algorithm, primality test by $O(\sqrt{N})$ trial division, Sieve of Eratosthenes, efficient exponentiation)</p> <p>Simple iterative algorithms (min/max selection, histogram, bucket sort)</p> <p>Sequential and binary search algorithms</p> <p>Search by elimination, "slope" search</p> <p>Quadratic sorting algorithms (selection, insertion)</p> <p>Partitioning, order statistics by repeated partitioning, Quicksort</p> <p>$O(N \log N)$ worst-case sorting algorithms (heap sort, merge sort)</p> <p>Binary search trees</p> <p>Representations of graphs (adjacency list, adjacency matrix)</p>
--	--	--	--	---	---

		<p>software. It is envisaged that work on the Project will begin in parallel with work on Section 3.</p>		<p>primary memory technologies, secondary storage devices and data communications in general, non-technical terms. Computer systems: Students should understand that an operating system is a collection of programs performing: Communication with peripherals Coordination of processes(including programs) Memory management File handling Accounting Security Data management Error handling Students should understand the need for systems software to provide: Utilities such as archiving (and compression), de-fragmentation, file maintenance A user interface (and be able to compare graphical user interfaces with command line interface) Data representation and graphics: The relationship between numbers of patterns and number of bits should be known, e.g. 24-bits allows 2^{24} or (approx) 16 million colours. The relationship between resolution and file size should also be understood and students should appreciate that different resolutions are appropriate for particular devices. Application skills</p>	<p>Traversals of ordered trees Depth- and breadth-first traversals of graphs Shortest-path algorithms (Dijkstra's and Floyd's algorithms) Transitive closure (Floyd's algorithm) Minimum spanning tree (Prim's and Kruskal's algorithms) Topological sort Algorithms to determine connected components of an undirected graph Algorithms to determine (existence of) an Euler path/cycle Advanced algorithmic analysis: Minimax algorithms for optimal game playing Geometric algorithms: (on 2D grids, i.e. integer (x, y)-coordinates) Line segments: properties, intersections Point location w.r.t. simple polygon Convex hull finding algorithms 8In terms of a disjoint-set ADT. Software Engineering In the IOI competition, the application of software engineering concerns the use of light-weight techniques for small, one-off, single-developer projects under time pressure. Software design: Fundamental design concepts and principles Design patterns Structured design Transform an abstract algorithm into a concrete, efficient program expressed in one of the allowed programming languages, possibly using standard or competition-specific libraries. Make their programs read data from and write data to text files according to a prescribed simple format. Using APIs: API programming In particular, contestants may be expected to Use competition-specific libraries according to the provided specification. Software tools and environments: Programming environments (incl. IDE) In particular, contestants may be expected to Write and edit program texts using one of the provided program editors. Compile and execute their own programs. Debug their own programs. Software processes: Software life-cycle and process models In particular, contestants may be expected to Understand the various phases in the solution development process and select appropriate approaches. Excluded: Process assessment models, Software process metrics Software requirements and specification: Functional and nonfunctional requirements Basic concepts of formal specification techniques In particular, contestants may be expected to Transform a precise natural-language description (with or without mathematical formalism) into a problem in terms of a computational model, including an understanding of the efficiency requirements. Software validation: Testing fundamentals, including test plan creation and test case generation Black-box and white-box testing techniques Unit, integration, validation, and system testing Inspections In particular, contestants may be expected to Apply techniques that maximize the the opportunity to detect common</p>
--	--	--	--	---	---

				<p>Spreadsheet usage: Cell formats Formulae Simple graph plotting</p>	<p>errors (e.g. through well-structured code, code review, built-in tests, test execution). Test (parts of) their own programs. Software project management: Project scheduling (especially time management) Risk analysis Software configuration management In particular, contestants may be expected to Manage time spent on various activities. Weigh risks when choosing between alternative approaches. Keep track of various versions and their status while developing solutions. Formal methods: Formal methods concepts (notion of correctness proof, invariant) Pre and post assertions In particular, contestants may be expected to Reason about the correctness and efficiency of algorithms and programs. Computer Literacy Contestants should know and understand the basic structure and operation of a computer (CPU, memory, I/O). They are expected to be able to use a standard computer with graphical user interface, its operating system with supporting applications, and the provided program development tools for the purpose of solving the competition tasks. In particular, some skill in file management is helpful (creating folders, copying and moving files). Details of these facilities will be stated in the Competition Rules of the particular IOI. Typically, some services are available through a standard web browser.</p>
--	--	--	--	--	---

