Contents

Membership of the CDC-HKEAA Committee on Information and Communication Technology (Senior Secondary)

(since December 2003)

Chairman:	Mr. NG Hok-ling
Vice-chairman:	Mr. WONG Kwong-wai, Raymond
Convener:	Senior Curriculum Development Officer, Education and Manpower Bureau (Mr. TANG Kin-hung, until 17 October 2004) (Mr. WU Man-wai, David, since 1 December 2004)
Members:	Dr. CHAN Fun-ting
	Mr. CHEUNG Kin-sun, Jackson
	Mr. CHEUNG Wah-sang
	Dr. FU Wai-chee, Ada
	Dr. LEE Kar-tin
	Ms. LEE Man-yee, Anna
	Dr. NG To-yee, Vincent
	Mr. TANG King-chi
	Mr. WONG Wing-hong, Robert
	Dr. YU Yuen-tak
	Subject Officer, Hong Kong Examinations and Assessment Authority (Mr. NG Ka-ming)
Co-opted Members:	Mr. WONG Kin-wai, Albert
	Ms. YING Evon
Secretary:	Curriculum Development Officer, Education and Manpower Bureau (Ms. CHUNG Sau-sim, Jessica)

Preamble

This Curriculum and Assessment Guide is one of the series prepared jointly by the Hong Kong Curriculum Development Council (CDC) and the Hong Kong Examinations and Assessment Authority (HKEAA). It forms the basis for learning and teaching of the subject curriculum as well as that for setting public assessment. The issue of this single document on curriculum and assessment aims at conveying a clear message to the public that assessment is an integral part of the school curriculum and at promoting the culture of "assessment for learning" to improve learning and teaching.

The CDC is an advisory body giving recommendations to the Hong Kong Special Administrative Region Government on all matters relating to curriculum development for the school system from kindergarten to sixth form. Its membership includes heads of schools, practicing teachers, parents, employers, academics from tertiary institutions, professionals from related fields or related bodies, representatives from the HKEAA and the Vocational Training Council, as well as officers from the Education and Manpower Bureau.

The HKEAA is an independent statutory body responsible for the conduct of the Hong Kong Certificate of Education Examination and the Hong Kong Advanced Level Examination. The governing council of the HKEAA includes members who are mainly drawn from the school sector, tertiary institutions and government bodies, professionals and persons experienced in commerce and industry.

This Curriculum and Assessment Guide is recommended by the Education and Manpower Bureau for use in secondary schools. The subject curriculum developed leads to the appropriate examination provided by the HKEAA. To this connection, the HKEAA has issued a handbook to provide information on the format of the public examination of the subject and the related rules and regulations.

The CDC and HKEAA will keep the subject curriculum under constant review and evaluation in the light of classroom experiences, students' performance in the public assessment, and the changing needs of society and students. All comments and suggestions on this Curriculum and Assessment Guide should be sent to:

Chief Curriculum Development Officer (Technology Education)
Curriculum Development Institute
Education and Manpower Bureau
5/F., 24 Tin Kwong Road
Kowloon, Hong Kong

I. Introduction

Advanced Level Computer Studies (the Curriculum hereafter) is recommended for use in Hong Kong secondary schools at S6-S7. It is a two-year course leading to the Hong Kong Advanced Level Examination and is targeted at all students with information technology skills at Level 3 of the IT Learning Targets (or S3 Computer Literacy level).

This is a revised edition of the previous curriculum published in 1998. The revision is overseen and monitored by the CDC-HKEAA Committee on Information and Communication Technology (Senior Secondary) which was established in 2003. There are two purposes behind the revision of the Curriculum:

i. Computer technologies are changing rapidly. The Curriculum should be updated and strengthened regarding the applications of computer systems, in order to facilitate students in future studies and/or to equip them with skills for migration into the workforce.

ii. With the newly developed S4-S5 Computer and Information Technology curriculum in 2003, there is a strong need to develop a curriculum which forms a part of the continuum in the learning of computer and related technologies from junior secondary through to university or work, and blends eventually into the Information and Communication Technology (ICT) curriculum in the New Senior Secondary Education.

In view of the latest developments of computer technologies and the purposes of the revision, the Curriculum has been reorganised, updated and revised with new modules on computer applications such as SOHO Networking added and obsolete topics such as Machine Logic and Assembly Language deleted or trimmed. The Curriculum is designed to provide as broad and as deep a study as possible, with intellectual standards suitable for students at Advanced Level.

This Curriculum and Assessment Guide serves as a framework to guide teachers on the contents and learning objectives which the Curriculum entails. It is presented with the overall aims and broad objectives, together with specific aims and objectives for each module of the Curriculum. The topics are accompanied by learning outcomes, which serve to further elaborate and clarify the learning experiences and the scope of the knowledge that students should attain. It also includes essential learning elements suggested for effective learning, which teachers may incorporate when designing learning and teaching activities. The scheme of assessment is included as supplementary information for teachers, paying particular attention to the area of School-based Assessment.

II. Aims and Objectives

Aims

The Curriculum is designed to provide students with

- knowledge, understanding and skills in the development and use of computer systems in a range of applications;
- an understanding of the organisation of computer systems and their underlying mechanisms to perform computational tasks;
- an appreciation of the impact on society arising from the rapid development of computer technologies; and
- an opportunity to develop skills in problem solving, communication, creativity and critical thinking, and learning to learn capabilities.

Objectives

Upon completion of this Curriculum, students should have developed

- knowledge and understanding of the organization and architecture of computer systems and the inter-relationship among hardware, software and data;
- knowledge and skills in using a range of applications software effectively, ethically and in a discriminatory manner to support information processing and problem solving;
- an understanding and experience in the ways that information is logically and sensibly organized, processed and manipulated by a computer;
- knowledge and skills in data communications and network development;
- an understanding of the systematic methods of problem formulation, solution design and selection, implementation of solutions, testing and documenting solutions to problems; and
- an understanding and appraisal of the social and ethical issues pertaining to computer technologies.

III. The Curriculum Framework

Organisation of the Curriculum

The curriculum is organised around six mandatory modules that relate to the fundamental concepts of computer organisation and networking, information processing using some common applications software, and the developmental processes in programs and systems. Together with School-based Assessment, the entire curriculum comprises these modules: Office and Internet Applications, Databases, SOHO Networking, Computer Organisation, Systems Development and Programming. The former three modules are common modules to Advanced Supplementary Level Computer Applications. For each module, the following time allocation is recommended:

	Module	**No. of hours allocated**	**No. of lessons allocated (40 minutes/lesson)**
1	Office & Internet Applications	34	51
2	Databases	32	48
3	SOHO Networking	32	48
4	Computer Organisation	38	57
5	Systems Development	21	32
6	Programming	58	87
	School-based Assessment	19	29
	Total Curriculum Time	**234 hours**	**352 lessons**

It is recommended that schools should allocate 8 lessons (40 minutes per lesson) per week, or 10 lessons per 6-day cycle, for the Curriculum. In addition, 29 lessons should be made available for School-based Assessment. A total of 352 lessons (234 hours) is recommended for the implementation of the Curriculum.

Details of the modules are given in subsequent pages in the following order:

i. Office and Internet Applications
ii. Databases
iii. Small Office / Home Office (SOHO) Networking
iv. Computer Organisation
v. Systems Development
vi. Programming

The order of the modules does not in any way dictate how the curriculum is to be implemented. The organisation of topics in each module represents one possible way of arranging the curriculum content. Teachers may structure and design teaching schemes according to their school situations, student needs, interests and abilities.

Office and Internet Applications

This module builds on the basic knowledge, understanding and skills of word processing, spreadsheet, multimedia presentation and Internet basics acquired by students at junior secondary or at S4-5 level. The aim of this module is to further develop students' understanding of terminology and advanced skills in using integrated software and the Internet, so as to enhance their personal productivity in work or study, and to apply them in problem solving. Emphasis is on the proficient use of software applications as well as the ability to use those applications as tools in enhancing document production, data analysis, decision-making, information management, communication and effective information presentation.

Overall Expectations

Students should

- understand the advanced features available in common word processing, spreadsheets and presentation software, and be able to integrate these features in a given context for problem solving;
- be able to select appropriate tools to process different types of information;
- understand the technology, applications and services of the Internet;
- have a basic understanding of some common multimedia file formats and be able to select the appropriate formats when presenting information; and
- be ethical and discriminating users of the Internet.

The time allocation for the module is approximately 34 hours (51 lessons).

Details of the Module

The module *'Office and Internet Applications'* comprises four topics; namely *'Using Word Processing in Desktop Publishing', 'Using Spreadsheet in Data Analysis', 'Multimedia Presentation of Information' and 'Internet Applications'*. Further information on the four topics is summarised as follows:

Topics	Remarks
A. Using Word Processing in Desktop Publishing	*(7 hours)*
• The design and production of formatted document / publication	• Students should demonstrate the ability to present information in a document suitably and effectively with features such as text frames, tables, multi-columns, section breaks, borders and graphics, etc., with due consideration on the use of colour, size and positioning of text and graphics. • Students should be able to create a document or a report proficiently with features such as table of contents, index, footnote / endnote, headers / footers, bookmarks and hyperlinks. They should be able to use tools to track changes and to enhance the readability and accuracy of documents using spelling checker, grammar checker and thesaurus. • Students should develop an understanding of the concepts of Object Linking and Embedding (OLE) and be able to compare the integration of graphics and data into the document by linking, embedding and copy/paste. • Students should recognise the different document / text formats such as plain text format, rich text format, hypertext document format, portable document format, and word document format. They should be able to do conversions among them, appreciate their uses and justify the choice of file formats in a given context. • Students should have experience of mail merging.
B. Using Spreadsheet in Data Analysis	*(12 hours)*
• The basic structure of a spreadsheet	• Students should have a clear understanding of the basic features of spreadsheets such as rows, columns, cell addresses and cell references, values, labels, formulas and charts, etc., and be able to apply them effectively in data analysis and data manipulation. They should demonstrate the ability to format or edit a worksheet efficiently.

Topics	Remarks
• Data manipulation	• Students should be able to use formulas, standard functions[1] and nested functions, together with mathematical, logical and relational operators, to solve problems. • Students should be able to arrange data in order, filter data using single or multiple criteria, link and manipulate data dynamically in multiple worksheets.
• "What-if" analysis	• Students should understand the use of spreadsheet in "what-if" analysis and appreciate how it is used to simulate some real world situations. They should be able to analyse and identify changes and trends so as to make informed judgment, decision and prediction when some values on the simulation are changed.
• Pivot tables	• Students should be able to create, edit and format a pivot table and a pivot chart efficiently using defined field names and appreciate the use of a pivot table as a powerful and interactive tool for data analysis. • Students should be able to use functions such as sum, sub-total and average, etc., and know how to group and manipulate data within pivot tables for problem solving. • By varying different fields, students should be able to observe and analyse the inter-dependency of data so as to enhance their decision-making and to produce meaningful predictions.

[1] Examples of functions are

Logical functions:	AND, OR, NOT, IF
Statistical functions:	AVG, MAX, MIN, LARGE, SMALL, SUM, SUMIF, COUNT, COUNTIF, RANK, FREQUENCY
Time:	DATE, NOW
Text functions:	CHAR, LEFT, LEN, LOWER, MID, RIGHT, UPPER, VALUE, CONCATENATE
Mathematical functions:	INT, MOD, ROUND, ROUNDUP, ROUNDDOWN, SQRT
Reference functions:	CHOOSE, HLOOKUP, VLOOKUP, LOOKUP

The list is by no means exhaustive. In essence, students should understand the basic construct of a function and know the parameters to be filled in for a function. They may need to use functions which are not on the list by referring to the HELP menu of the software when required.

Topics	Remarks
C. Multimedia Presentation of Information	*(4 hours)*
- Multimedia elements	- Students should have a basic understanding of the features of the different multimedia elements comprising video, audio, text and graphics. They should be able to use them, justify their usages, perform simple file conversions and compare the different file formats in terms of resolution, file size and applications.
- Multimedia presentation	- Students should be able to apply concepts for effective communication and presentation, and to enhance information presentation using multimedia elements.
D. Internet Applications	*(11 hours)*
- Internet Basics	- Students should be able to describe the hardware and software requirements for Internet access. - Students should know how data is transmitted over the Internet and understand concepts of Internet Protocol (IP), Uniform Resource Locator (URL), Domain Name System (DNS) and Hypertext Transfer Protocol (HTTP).
- Internet services and resources	- Students should be able to formulate an effective strategy for searching for specific information on the World Wide Web by using search-engines, and be able to critically analyse the sources of information. - Students should have experience of the use of the Internet for file transfer by using file transfer protocol (FTP), remote logon, locating and using an online chat, joining discussion forums and newsgroups. - Students should be able to use plug-ins or players for multimedia elements found on the Internet. - Students should understand the differences between a mail client and web mail, and the protocols (POP, IMAP, SMTP, etc.) used in sending and retrieving emails.
- Ethical and social issues on the use of the Internet	- Students should be able to discuss critically issues arising from the digital divide, the emergence of a knowledge-based society and globalisation. - Students should be keenly aware of the issues of intellectual property, and be familiar with measures to safeguard themselves and their information on computers, by various means such as the installation of a firewall, filtering software, anti-spyware and anti-virus software. They should understand the use of cookies and the impact on user tracking. - Students should understand the measures which ensure Internet security in data transmission, such as the use of a digital certificates and data encryption.

Databases

With the widespread use of databases within our society, everyone now needs knowledge and skills in database management in order to make meaningful use of this tool. This module introduces students to the fundamentals of databases and relational database design. It entails the application, management and design aspects of databases. Students will learn how to construct simple data models using entity-relationship diagrams and to appreciate the importance of good database design. They will also learn to use Structured Query Language (SQL) to construct, manipulate and retrieve information from a relational database. Through the completion of this module, students will acquire a basic understanding of the concepts, skills and applications of databases, and elementary data modeling concepts.

Overall Expectations

Students should

- be able to explain concepts and applications related to databases and the database management system (DBMS);
- understand the basic concepts of a relational database and be able to construct, manipulate and extract information from a relational database using Structured Query Language;
- be able to identify and perform analysis of the data requirements of simple business systems;
- be able to construct simple data models using methodologies such as entity-relationship (ER) diagrams; and
- appreciate the importance of a good database design as a blueprint for the development of a database system.

The time allocation for the module is about 32 hours (48 lessons).

Details of the Module

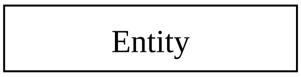
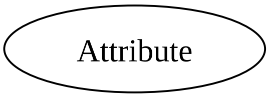
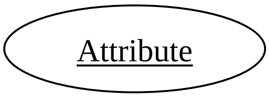
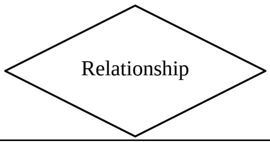
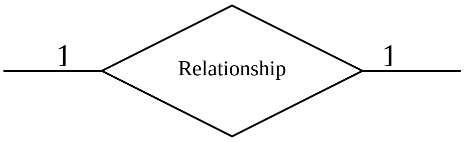
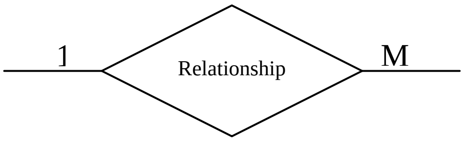
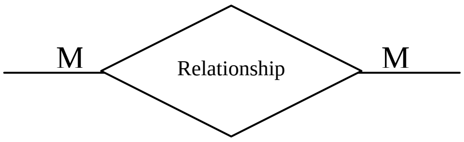
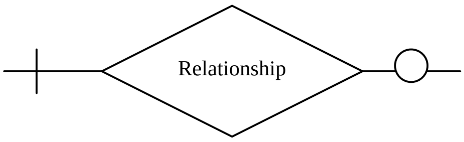
The module *'Databases'* comprises three topics; namely *'Introduction to Databases'*, *'Relational Databases and Working with Structured Query Language (SQL)'* and *'Introduction to Database Design Methodology'*. Further information on the three topics is summarised as follows:

Topics	Remarks
A. Introduction to Databases	*(3 hours)*
• Applications of databases in society	• Students should be aware of the uses and applications of databases in everyday life (e.g. the library system, inventory system in a supermarket, credit card system, etc.). • Students should be given opportunities to discuss the importance of databases in business environments and how they are related to the success of a business.
• Concepts and terminology	• Students should understand the following terminology and concepts: ➢ data and information ➢ data, fields, records, tables, files and databases ➢ common data types such as integer, real, character, string, Boolean, date, etc. ➢ indexes and keys ➢ database management systems (e.g. data definition language, data manipulation language, data dictionary, transaction processing and access control, etc.) ➢ program-data independence ➢ data redundancy and data integrity
B. Relational Databases and Working with Structured Query Language	*(18 hours)*
• Basic concepts of a relational database	• Students should know the basic concepts underpinning relational databases such as entity, relation, attribute, domain, primary key, foreign key, candidate key, entity integrity, referential integrity, domain integrity, etc. Students should be able to identify these basic elements in examples taken from everyday applications. • Students should know how to organise data differently but sensibly in a relational database and be able to establish the required relationships to link up the tables.
• Creating a relational database	• Students should be able to create a simple relational database[2] based on specified requirements using SQL.

[2] A simple relational database refers to one working with at most 3 tables.

Topics	Remarks
• Database maintenance and manipulation	• Students should be able to use SQL to maintain a simple relational database, manipulate its data or retrieve the required information. They should be able to: ➢ modify the structure of the tables ➢ add, delete and modify the data in the tables ➢ view, sort and select the contents by filtering ➢ use appropriate operators and expressions such as the *in*, *between* and *like* operators, arithmetic operators and expressions, comparison operators and logical operators etc. to perform specific operations ➢ use simple built-in functions such as aggregate and string functions, etc. ➢ perform multiple field indexing and multi-level ordering ➢ perform queries on multiple tables including the use of equi-join, natural join and outer join ➢ perform sub-queries (for 1 sub-level only) ➢ export query results to, for example, text, html or spreadsheet format, etc.
C. Introduction to Database Design Methodology	*(11 hours)*
• The conceptual data model	• Students should understand the importance of good database design in effective database management. They should be aware of the three levels of data abstraction; namely conceptual level, physical level and view level.
• Entity-Relationship modeling	• Students should be aware of the three types of relationship (one-to-one, one-to-many, many-to-many) among entities in a relational database. • Students should be able to create simple entity-relationship (ER) diagrams[3] involving binary relationships only in designing databases for simple business scenarios. This includes the resolution of many-to-many relationships into multiple one-to-many relationships in order to implement the database. • Students should be able to transform the ER diagrams to tables in relational databases and be able to create a database schema for a given set of data to describe the characteristics of the database.
• Introduction to Normalisation	• Students should be able to briefly explain the meaning and purpose of normalisation. They should be aware of the methods or measures used to reduce data redundancy.

[3] Symbols used in entity – relationship diagrams:

Meaning	Symbol
Entity	
Attribute	
Key Attribute	
Relationship	
One-to-One Relationship	
One -to-Many Relationship	
Many-to-Many Relationship	
Participation constraints: • Use | on Mandatory side • Use ○ on Optional side	

Small Office/Home Office (SOHO) Networking

This module focuses on knowledge and skills essential in constructing SOHO networks. SOHO networks are small-scale networks suitable for small-and-medium enterprises (SMEs) which make up more than 90% of businesses in Hong Kong. Building up networking capabilities and infrastructure within SMEs will increase the productivity and enhance communication internally among staff and externally to customers or clients throughout the world.

Through the study of this module, students will gain an understanding of the basic principles of networking, and the knowledge and skills associated with the design, implementation and maintenance of a SOHO network. Students will also be aware of the common security threats to SOHO networks and the methods by which network security can be improved. This module will give students a solid foundation on networking in general and SOHO networking in particular. Students will appreciate the practical nature of the module, regardless of whether they continue to pursue knowledge in this area in tertiary education or migrate to the workforce.

Overall Expectations

Students should

- understand the basic concepts and technologies behind networking, focusing specifically on SOHO networks;
- be able to identify and describe the functions of basic components involved in a simple network;
- be able to describe the uses and applications of a SOHO network;
- be able to design and implement a simple SOHO network;
- be able to assess the performance of a network and implement improvements; and
- understand the importance of network security and be able to propose measures of improvement.

The time allocation for the module is about 32 hours (48 lessons).

Details of the Module

The module *'Small Office/Home Office (SOHO) Networking'* comprises three topics; namely *'SOHO Networking Basics'*, *'SOHO Network Design and Implementation'* and *'SOHO Network Management and Security*'. Further information on the three topics is summarised as follows:

Topics	Remarks
A. SOHO Networking Basics	*(8 hours)*
• Basic concepts of data communications and networking	• Students should be able to identify and describe the basic function of each component of a packet: header, data and trailer. They should also be able to explain briefly the use of packets in data transfer in a packet switching network. • Students should understand the basic concepts of Internet Protocol (IP) addressing including the schemes and classification of IP addresses. They should understand the use of a subnet and know which subnet an IP address belongs to from a simple subnet mask. • Students should know the use of some common protocols including TCP/IP and DHCP, etc. • Students should be able to explain the factors that need to be considered in choosing between a client-server network and a peer-to-peer network. • Students should be able to compare and contrast the common types of communication links (e.g. modem dialup or cable modem, leased line, broadband and wireless, etc.) for Internet access in terms of data transfer rate, cost, and reliability.
• Basic network components	• Students should be able to identify and describe the functions of the various components which make up the wired and wireless networks. These include the network interface cards (NICs), cables, hubs, switches, routers, broadband routers, gateways, wireless adapters, wireless access points, wireless routers, etc. They should also be able to describe and explain briefly the services provided by a network operating system.
• SOHO network applications	• Students should be able to describe and appreciate the common applications of SOHO networking including resources sharing, Internet access, web serving, telecommuting, etc.

Topics	Remarks
B. SOHO Network Design and Implementation	*(16 hours)*
• Need analysis	• Students should be able to conduct a simple need analysis on a proposed SOHO network and translate the needs identified into requirements and specifications.
• Design	• Students should be able to design a network to meet the requirements generated in the need analysis and represent it in a diagram. • Students should be able to justify their design based on technical, cost-effectiveness and other considerations.
• Setup	• Students should have experience of setting up simple Ethernet and wireless networks. • Students should have experience of sharing various resources (e.g. files, printers and Internet connection, etc.) among the networked computers/stations. • Students should have experience of setting folder/file-sharing permissions including read, write and execute rights, etc.
• Testing	• Students should have experience of validating a network system by testing it according to a simple test plan based on the requirements and specifications.
• Documentation	• Students should be able to document the user requirements, specifications, and a schematic diagram for the network.
• End-user support	• Students should be aware of the importance of adequate end-user support and training on the attainment of the benefits sought.

Topics	Remarks
C. SOHO Network Management and Security	*(8 hours)*
• Monitoring, fine-tuning and troubleshooting	• Students should develop the basic skills of monitoring and fine-tuning the performance of a simple network. • Students should develop the basic skills of analysing problems associated with the use of a networked environment and performing troubleshooting for it.
• Backup	• Students should be aware of the importance of backup in disaster planning and recovery measures. • Student should know the common hardware and software components of a network backup solution, such as Redundant Array of Independent Disks (RAID), Uninterruptible Power Supply (UPS), network backup servers, network backup and recovery software, etc.
• Security threats and measures	• Students should be able to describe the potential risks caused by the common network security threats including virus, worm and Trojan programs, spyware, unauthorised access, interception, etc. • Students should be able to propose effective measures to improve network security for both wired and wireless networks. These include anti-virus programs, authentication, access and user right control, packet filtering, public and private key encryption, Wired Equivalent Privacy (WEP), and IPsec used in Virtual Private Network (VPN), etc.

Computer Organization

This module is designed to provide students with the basic concepts of the architecture of a computer and how it works. These concepts include the learning of the functional units of a computer, the use of buses, main memory, peripherals and the various forms of data representation and manipulation within a computer system. Through the learning of elementary assembly language and the methods by which data and instructions are fetched, decoded and executed, the module provides students with a basic understanding of the interaction of various components in a low-level run-time environment. In addition, students will be exposed to several alternative designs for CPUs and the concepts employed in the design of high-performance computer systems.

Overall Expectations

Students should

- identify and explain the functions of the major components of a computer system, such as the CPU, storage devices and peripherals etc., and how these components interact together to perform tasks;
- identify and compare the different bus systems;
- know how data and instructions are represented inside a computer, and how they are fetched, decoded, and executed within a computer system;
- have a basic understanding of the capabilities of the operating system;
- appreciate the alternative designs for CPUs; and
- appraise the effects of architecture and organisation of computer on computer performance.

The time allocation for the module is about 38 hours (57 lessons).

Details of the Module

The module *'Computer Organization'* comprises five topics; namely *'Basic Machine Organisation'*, *'Basic Principles of Assembly Language'*, *'Data Representation and Manipulation'* , *'Operating Systems'* and *'Modern Computer Architectures'*. Further information on the five topics is summarised as follows:

Topics	Remarks
A. Basic Machine Organization	*(11 hours)*
• Basic concepts of a computer system	• Students should understand the basic concepts such as input-process-output cycle and the use of a stored program in a processing system. • Students should be able to identify the basic components of a computer system and know the general functions of hardware and software.
• Central Processing Unit	• Students should be able to explain the functions and characteristics of the Central Processing Unit (CPU), the Control Unit (CU) and the Arithmetic and Logic Unit (ALU). • Students should be able to give an account of the fetch-decode-execute cycle to include the automatic execution of the sequences of instructions and the inter-dependency among components, registers and buses within the system. • Students should understand the functions and the inter-relationship of the following registers in a machine cycle: program counter, accumulator, instruction register, program status register, general purpose registers, memory address register and memory data register. • Students should be able to describe the functions of the different types of buses and understand the importance of synchronization of data transfer along buses.
• Main Memory	• Students should be able to identify the functional characteristics of immediate access storage such as RAM, ROM, PROM and EPROM, and be able to describe the logical organisation of the memory system using memory chips. • Students should be able to describe the relationship between the size of the memory, the memory address, word length and the performance of the computer.
• Peripheral devices	• Students should be able to explain the functional characteristics of peripheral devices (I/O devices and backing store). They should be able to identify the associated media of the peripherals and discuss their data access time and transfer rate.

Topics	Remarks
• Peripheral handlings	• Students should be aware of the measures used by CPUs in handling peripheral requests such as polling and interrupts. • Students should understand the various ways of controlling data transmission to and from peripherals and the related concepts such as the use of direct memory access (DMA) and buffers including double buffering technique.
B. Basic Principles of Assembly Language	*(10 hours)*
• Instruction formats	• Students should understand the basic format of a machine code instruction and the meanings of operation codes and operands. • Students should understand the purposes of using assembly language. They should be able to explain the meanings and uses of mnemonic operation codes and symbolic addressing, and describe the format of assembly language instruction and its relationship to machine code.
• Modes of addressing	• Students should understand the different types of addressing modes: direct, indirect and immediate.
• Basic assembly language instructions	• Students should understand the data transfer instructions such as move, load and store. • Students should be able to use the arithmetic and logical instructions of ADD, SUBTRACT, AND, OR, NOT, XOR, INCREMENT and DECREMENT. • Students should understand the concepts and uses of conditional jump and unconditional jump in the transfer of control instructions.
C. Data Representation and Manipulation	*(9 hours)*
• Numeric data representation	• Students should understand the representation of numbers in binary, and their equivalent in denary and hexadecimal number systems, or vice versa. • Students should understand how integers are represented using the two’s complement and sign-and-magnitude methods. They should be able to perform addition and subtraction operations in binary. • Students should know the representation of real numbers by fixed-point representation and IEEE 32-bit floating point representation. They should also develop an understanding of precision and range, and the use of hidden bit to improve precision and multiple precision representations of floating point numbers. • Students should be able to analyse errors arising from the operations of truncation, rounding, overflow and underflow.

Topics	Remarks
⬩ Non-numeric data representation	⬩ Students should have a basic understanding of the representation of characters using common international standards such as the ASCII code, the Big-5 code and the Unicode. They should be able to describe the relationship between the size of the character set and the representation.
D. Operating Systems	*(4 hours)*
⬩ Functions of operating systems	⬩ Students should understand what system software is and be able to give a brief account of the basic functions of an operating system. ⬩ Students should know the relationship between the hardware, system software, application software and users.
⬩ Types of operating systems	⬩ Students should be able to distinguish between the different types of operating systems as listed below. They should also be able to understand the limits and benefits of the various types of operating systems. ➢ Batch processing systems ➢ Multitasking and multiprogramming systems ➢ Online interactive and real-time systems ➢ Single-user and multi-users systems ➢ Network systems
E. Modern Computer Architectures	*(4 hours)*
⬩ Limitations of traditional model	⬩ Students should have the opportunity to appraise the Von Neumann model and explore possible measures such as the use of cache memory to enhance the performance of a computer system.
⬩ Alternative architectures	⬩ Students should demonstrate a basic understanding of the following alternative architectures as means to raise the performance of a computer system: ➢ Parallel and distributed processing ➢ Pipelining ➢ Complex Instruction Set Computer (CISC) ➢ Reduced Instruction Set Computer (RISC)

Systems Development

This module introduces students to the main principles of systems development which is in fact a systems approach to problem solving. Through the study of this module students will acquire, in a sequential manner, the concepts of systems, the purposes of systems development, the formal procedures or phases, methods, tools and systems specifications involved in developing a new system. Students will also be exposed to some of the latest approaches in systems development.

The knowledge, skills and methodologies involved in this module help to develop students' systems thinking. This enables students to see concepts as complete entities, not just fragmented ideas, and to observe interrelationships among components rather than discrete things. This leads to an understanding of the intricate networks of related activities and events within an information system. This kind of ability, crucial in project management, is a skill with which students should be equipped.

Overall Expectations

Students should

- understand the concepts of a system;
- develop abilities in analysing a system, defining problems and formulating computer-based solutions;
- identify the main phases involved in systems development;
- describe the main activities, processes and methodologies used for systems development;
- use simple charts, diagrams or other tools to document a system development process; and
- appreciate the importance of a systemic approach to solve problems and develop the techniques appropriate for implementing such solutions.

The time allocation for the module is about 21 hours (32 lessons).

Details of the Module

The module *'Systems Development'* comprises three topics; namely *'Basic Concepts of Systems Development'*, *'Phases of Systems Development' and 'Alternative Approaches to Systems Development'*. Further information on the three topics is summarised as follows:

Topics	Remarks
A. Basic Concepts of Systems Development	*(2 hours)*
• Concepts of a system	• Students should understand what a system is in general, and an information system in particular, and be able to identify its basic elements such as environment, problem domain, inputs, outputs, processes, interfaces and storage, when put in a given context. • Students should be aware of the merits and drawbacks of developing a new system and the main phases involved in systems development using the Waterfall Model.
B. Phases of Systems Development	*(16 hours)*
• Systems Analysis	• Students should have a basic understanding of some common methods / activities such as interviews, surveys, questionnaires, observations and document reviews in order to gather information for analysing problems, identifying and documenting users' requirements. • Students should understand the need and use of a feasibility study to determine whether the proposed solution is feasible. • Students should have the opportunity to formulate and evaluate alternative proposals and to develop requirement specifications.
• Systems Design	• Students should have the experience to specify and document the functions of each part of a computer-based system including the hardware platform, software, inputs and outputs, user interface, database structure, process and control on data and security, etc. • Students should have the experience to design and appraise an effective interface between the user and the system, and be able to compare the different types of human-machine interface such as command-driven interface, menu-driven interface and graphical user interface. • Students should be able to design, analyse and document a proposed system by using methodologies such as system flowchart, structure chart, data flow diagram and data dictionary. • Students should understand the importance of project planning. They should be able to use critical path to analyse the time required, to allocate resources and to plan the breakdown of the project work.

Topics	Remarks
- Systems Implementation	- Students should be able to identify the major tasks involved in this phase of systems development. They should be able to design, create and develop computer-based solutions, and be able to document the process. - Students should be able to explain the different types of testing (unit testing, system testing and acceptance testing) and their uses, and be able to devise a simple test plan or test procedure of a system, or part of a system, to evaluate its actual functionality.
- Systems Conversion and Maintenance	- Students should be able to compare the different strategies of systems introduction / systems conversion, such as pilot conversion, phased conversion, parallel conversion and direct conversion, and be able to explain the needs for on-going maintenance and user trainings.
- Systems Documentation	- Students should know that documentation is an on-going process and is integrated into various stages of systems development. - Students should have experience of documenting the requirement reports, project plans or design plans, etc. - Students should be able to differentiate between types of documentations (system documentation, technical documentation and user manual) and be aware of the purposes of each of them.
C. Alternative Approaches in Systems Development	*(3 hours)*
- Other approaches	- Students should be given the opportunity to explore and analyse the limitations and shortcomings of the Waterfall Model in systems development. - Students should develop an understanding of the benefits and limitations of some of the latest approaches such as prototying and rapid application development (RAD). They should be able to compare the various types of approaches and be able to suggest with justifications the most appropriate approach in developing a system in a given context.

Programming

The aim of the module is two-fold: firstly to introduce students to programming concepts and programming languages, and to develop students problem solving skills through a systematic approach to algorithm design and programming.

Through the study of this module students will understand the whole process of developing a program, the computer-based solution, as a means to solve a problem. In designing a solution, students will acquire various principles and techniques of algorithm design as well as the different data types and data structures, all of which will facilitate the construction of a program. The programming languages, which are used to realise the solution in mind, can be one of the following: Pascal, C, Visual Basic or Java. In the latter part of the module, students will be exposed to different programming paradigms and will acquire concepts of the programming languages associated with these programming paradigms at an introductory level. This may raise students' awareness in various programming languages and understanding of the suitability of certain programming languages in solving specific types of problem.

Overall Expectations

Students should

- have a basic understanding of the program development process and be able to describe the steps involved in developing a program;
- acquire the skills in developing and writing programs including the skills in defining a problem, analysing it, designing and formulating a possible solution, debugging and testing the solution, as well as documenting the whole process;
- understand the principles and techniques of algorithm selection and design, and be able to apply them;
- understand the concepts and uses of data types and data structures, and be able to apply them in solving computer-based problems;
- be aware of the development of programming languages and their corresponding programming paradigms; and
- develop an appreciation of the use of different languages for meeting different needs.

The time allocation for the elective is about 58 hours (87 lessons).

Details of the Module

The module *'Programming'* comprises two topics; namely *'Programming'* and *'Programming Languages'*. Further information on the two topics is summarised as follows:

Topics	Remarks
A. Programming	*(49 hours)*
• Introduction	• Students should understand that program development is a cyclical process. They should be able to describe and explain the stages commonly employed in developing programs.
• Problem Definition and Analysis	• Students should be able to define the problems and identify the necessary inputs, expected outputs and tasks involved in a given problem.
• Planning and Design of Solution: *Planning Approach*	• Students should be able to apply top-down and bottom-up problem-solving approaches in developing programs. They should be able to explain concepts of modularity, stepwise refinement and structured programming.
• Planning and Design of Solution: *Data Type and Data Structure*	• Students should know the need for different data types to reflect the data requirements of a problem or application. They should understand the nature of simple data types, structured data types and user-defined data types. • Students should know how and why data are organized in lists, linear linked lists, stacks and queues, and be able to differentiate their uses and differences. • Students should understand the concepts of binary search trees and their traversals, and be able to represent them in simple diagrams.
• Planning and Design of Solution: *Algorithms*	• Students should be able to derive and describe an algorithm in pseudocode and flowchart for a given problem. • Students should be able to describe, represent, test and analyse the following algorithms: algorithms for ➢ selection, iteration and recursion ➢ counting, accumulating and swapping ➢ accessing and updating data structures and files ➢ sorting including bubble sort, insertion sort, quick sort and merge sort ➢ merging two arrays of data ➢ searching including linear search and binary search • Students should be aware that a problem may be solved by several algorithms and that selection of algorithms sometimes depends on their complexities and storage requirements as well as their trade-off.

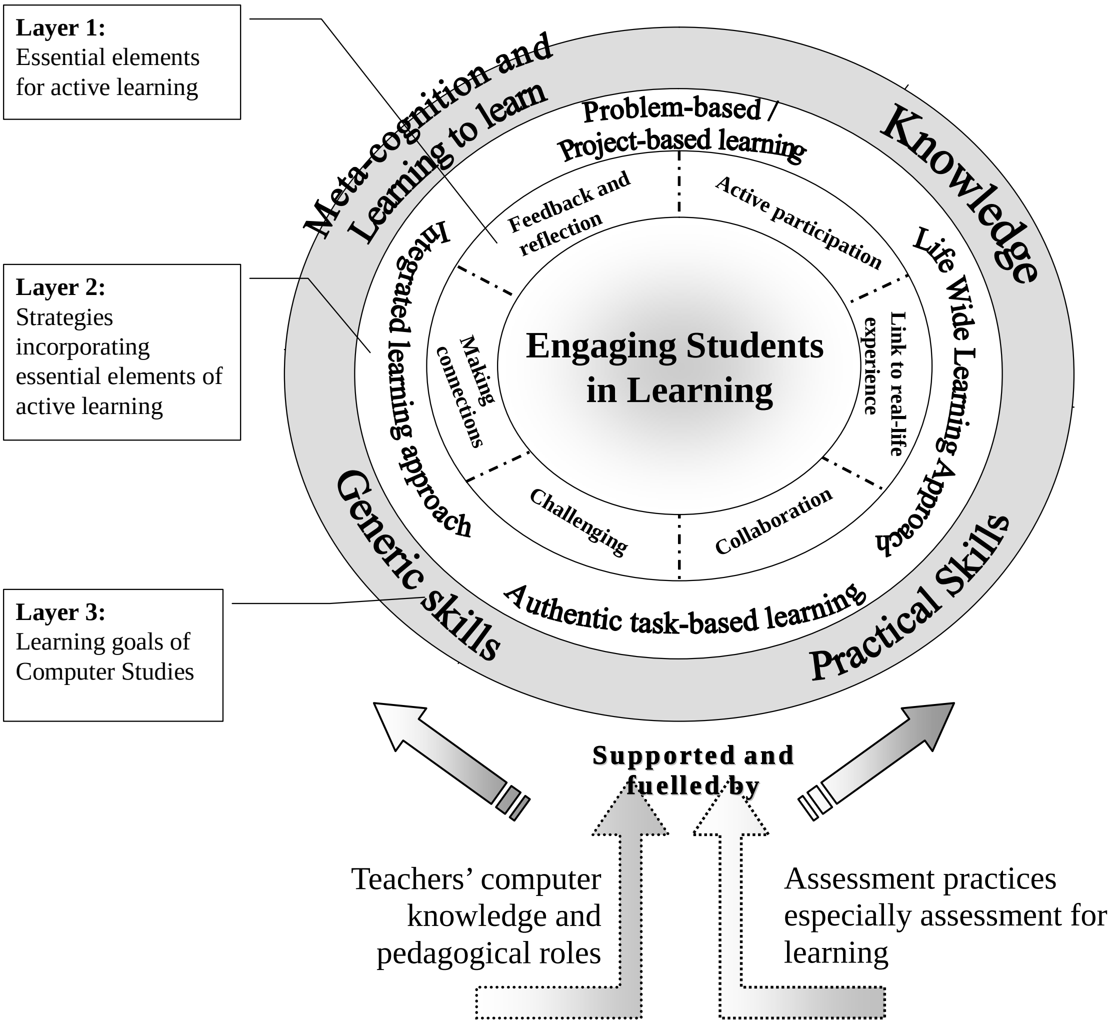
Topics	Remarks
• Implementation	*(In implementing the program, based on teachers' expertise and school's resources, teachers may teach one of the following languages; Pascal, C, Visual Basic or Java.)* • Students should be able to distinguish declarative and executable statements. They should be able to map the data types and algorithms to the chosen programming language. • Students should understand the language features of the selected programming language: ➢ the definition of data types and manipulation of data structures (lists, linear linked lists, stacks and queues only) using arrays. ➢ basic programming constructs such as constants and variables, expressions and statements (input, output and assignment statements), various operators (arithmetic, string and Boolean operators) and operations, control structures (sequence, selection and iteration). ➢ file handling operations (for text files only) including file updating statements to delete, insert, append and amend records. ➢ subprograms with parameters passing (call by value and call by reference) and recursion. • Students should appreciate the use of a modular structure to design, implement and identify debugging errors.
• Testing and Evaluation	• Students should be able to trace and debug programs by manual methods or by use of software debugging tools such as stubs, flags, break points, etc. They should be able to design a set of test data for program testing, and be able to identify and correct syntax, logical and run-time errors.
• Documentation	• Students should develop the habit of documenting the processes of program development and be able to produce various forms of documents such as algorithm, program listing, test plan with tested data and user manual, etc.
B. Programming Languages	*(9 hours)*
• Evaluation of programming languages	• Students should develop a basic understanding of the programming paradigms involved in procedural, declarative, object-oriented, visual and query languages and be able to describe the criteria for selecting a programming language for problem-solving in a given context.
• Program translation and stages of compilation	• Students should have a basic understanding of the need for high-level programming languages and the processes involved in the translation and analysis of such programs. Concepts include lexical, syntactic and semantic analysis, code generation, linkers and loaders, and compilers and interpreters.

IV. Learning and Teaching

As the 21st Century is unfolded, there is a fundamental change in learning and teaching. The new culture of learning and teaching has shifted from transmission of knowledge to learning how to learn, from over-emphasising content study to whole person development, from compartmentalised topics / subjects to integrated learning, and from the dependence on textbooks to diversified teaching and learning materials. All these changes recognise the fact that citizens in the modern world need to be critical thinkers, problem solvers and competent in articulating ideas. Above all they need to be active learners who can construct knowledge taken from various sources and from different perspectives in a rapidly changing environment.

Every student has a capacity to learn. Students learn best when they actively seek information to accomplish a learning task. They learn effectively when they play the role of researchers and navigators rather than spectators; the role of problem solvers and thinkers rather than passive recipients of a static set of facts, as these facts may become obsolete as the rate of computer development accelerates. Students need to construct their own knowledge, learn from different perspectives and be responsible for their own learning. This paradigm of learning comes naturally and effectively if students are intrinsically motivated and engaged in their learning, i.e. they themselves become active learners.

To engage students actively in learning, as shown in the diagram overleaf, teachers should put them in the centre and blend the essential elements (Layer 1) required in effective learning into classroom learning and teaching strategies which are suggested in Layer 2. In addition to the learning of the Curriculum delivered via various student-centered learning and teaching strategies, and apart from knowledge and practical skills associated with computer technologies, students will develop generic skills such as problem solving, critical thinking, creativity and communication, and metacognition and learning to learn capabilities (Layer 3). These are all beneficial to students in the complex world of work and study. This system of learning should be supported by the different and active roles played by teachers and supported by the new concept of assessment, assessment for learning, to further facilitate student learning.



Layer 1:
Essential elements for active learning
Layer 2:
Strategies incorporating essential elements of active learning
Layer 3:
Learning goals of Computer Studies
Meta-cognition and Learning to learn
Knowledge
Generic skills
Practical Skills
Problem-based / Project-based learning
Life Wide Learning Approach
Authentic task-based learning
Integrated learning approach
Feedback and reflection
Active participation
Link to real-life experience
Collaboration
Challenging
Making connections
Engaging Students in Learning
Supported and fuelled by
Teachers' computer knowledge and pedagogical roles
Assessment practices especially assessment for learning

The System of Active Learning in A-level Computer Studies

Essential Elements for Active Learning

Learning occurs best

- when students create their knowledge actively using their own learning strategy, and engage in an active search for understanding of what is being learnt;
- when students make connections between concepts, skill elements and experiences;
- when it is structured in real-world problems or circumstances;
- in the context of relatively challenging tasks or problems;
- in a collaborative context that provides thought-provoking discussion, trust and supportive interaction from peers and teachers; and
- when students receive qualitative feedback, encouragement for self reflection and opportunity for practice and improvement.

Learning and Teaching Approaches

Learning in Computer Studies is a complex, multi-faceted, active and interactive process. Apart from the traditional lecturing approach, active learning elements can be infused into classroom activities for the effective learning and teaching of the Curriculum. The following approaches are suggested for teachers' reference only and are by no means the only way to deliver the Curriculum. Teachers may use an extensive repertoire of learning and teaching strategies to achieve the learning objectives of the Curriculum.

Task-based Learning: The task should be a goal-oriented activity with a clear purpose in mind. Teachers may provide learning opportunities that engage students actively in tasks that are related to real-world problems or circumstances. Students should be asked to perform, create, produce, or do something that invokes real world applications; for instance, the creation of an advertisement flier, the simulation of a tax payment, the creation of a school library database system or the setting up of a home network for shared internet access. The tasks, however, should be moderately challenging and interesting enough to increase students' motivation. Teachers need to have very clear teaching objectives. The tasks, however, are tools which serve as instruments to facilitate student acquisition of underpinning concepts and skills, not the learning objectives themselves.

Life Wide Learning Approach: To enrich students with real-life experience, activities such as visits to computer or IT companies or institutes help to widen their horizons on the applications of computer systems. Students interact with the environment with the purpose of exploring, learning and observing how problems are solved in a computer-based system. By giving students authentic experiences outside the school setting in the form of observation or problem-solving, which they may inevitably encounter in their own workplaces or study, deeper understanding and learning skills will be fostered in students. The inclusion of reflection on experience after visits can also maximize learning.

If visits or field experiences of this kind cannot be provided, case studies of real-life examples detailing how computers are employed in various contexts can be used to bring out the related concepts and methodologies and to further consolidate students' concepts, knowledge and skills.

Problem-based Learning and Project-based Learning: Most students retain and use little of what they memorise in classroom. Students learn best when they are actively and purposely seeking information.

> *"Tell me, and I will forget. Show me, and I may remember. Involve me, and I will understand." (Confucius around 450 BC)*

By engaging students in structuring solutions to real life and contextualized problems, problem-based learning or project-based learning helps to orient students towards knowledge-making over fact-collecting. Through searching for and finding solutions to problems, students develop higher levels of comprehension and cognitive strategies to research, gain more learning and knowledge-forming skills and more social skills (collaboration and handling group dynamics) if group work is involved. This kind of approach emphasizes long-term, integrated and student-centred practices. Students assume greater responsibility for their own learning due to the reduction in direct teaching.

As a result of their rich learning outcomes, problem-based learning and project work should be integrated throughout the Curriculum and embedded in learning and teaching activities, and the assessments. This kind of methodology is commonly used throughout the computer industry and students will emulate these approaches. The importance of this kind of learning is also reflected on the mandatory Project Assignment in School-based Assessment. However, problem-based or project-based learning should not be restricted to a Project Assignment only. If problem-based or project-based learning is structured and guided carefully with built-in mechanisms for self-reflection, learning in both knowledge and skills will be maximised.

Making Connections and Integrated Approach: The Curriculum should not be viewed as a collection of discrete and disconnected modules. Concepts, skill elements and experiences can be linked and integrated to make them more meaningful and achieve lasting cognitive connections. The learning of Systems Development is enhanced, for instance, if various stages of the development of systems can be connected, where appropriate, to other modules of the course in order to consolidate students' understanding of the processes or methodologies used in a particular stage. This may serve as a connector, linking concepts and skills acquired throughout the Curriculum into a holistic picture of information systems. By helping students to make connections they will be able to see the interrelationships between concepts within or beyond the Curriculum. This can facilitate flexible thinking, critical thinking and transfer of knowledge from one context to another.

Approach to Sustain Computer Fluency: Faced with the rapid advancement of computer technologies, students should be encouraged to read computer journals and magazines, to update their computer knowledge and skills through knowledge networks such as educational newsgroups and websites available on the Internet, and to participate in various workshops or contests in order to enrich their learning experiences. All of these help to stretch students' potential in computer knowledge and nurtures them into lifelong learners. In this respect they are similar to personnel working in IT industries, who continually keep abreast of new developments in the field.

Collaborative Approach: Effective learning is social and interactive. In the learning of computer and related technologies, students are often required to undertake group work. It is recommended that students at various ability levels should be grouped together in small groups when solving a problem or accomplishing a project or a task. This will enhance the active exchange of ideas and multiple feedback within the small groups. This not only increases interest and retention of information by students, but also promotes critical thinking, enhances communication skills and immediate feedback from peers. The shared learning gives students an opportunity to engage in discussion and take responsibility for their own learning.

Feedback and Reflection: Feedback from peers and teachers throughout the learning process or activity, be it a project, a task or a problem to solve, is essential for effective learning and should be integrated into learning and teaching. Structured opportunity for practice should be given, for without opportunity for practice, even well-learned abilities will disappear.

Reflection is necessary to reach the point of deeper learning required for knowledge and skills to be retained for use in the future. Reflection enhances students' self-assessment skills. Through reflection students use critical thinking to examine their understanding of the concepts, application of certain computer skills and presentation of work. They take control of their own learning. In doing so, students develop their learning to learn capabilities and metacognition. Without reflection, deep learning will not occur.

Effective learning can be achieved with the active involvement of teachers who can direct and orchestrate learning and teaching activities for students, and with a clear vision of assessments (especially assessment for learning) within and beyond the classroom.

Teachers’ Roles

New concepts of learning demand new concepts of teaching and different roles for teachers. To foster active learning teachers must not only be knowledgeable about computer technologies but also need to have the pedagogical skills and knowledge to deliver the Curriculum. First and foremost the teacher becomes a facilitator who may become the consultant or instructor of computer knowledge, resource guide or developer, assessor and learner in the learning community.

(i) Teachers are the *consultants* or *instructors* of computer knowledge, for they

- have a thorough understanding of the theories, principles and concepts of computer technologies and a flexible approach so that they can help students to create useful cognitive maps, construct knowledge, apply that knowledge to real-world settings, relate one idea to another and address misconceptions;
- serve as role models when they fulfill their roles as teachers. To inspire students with their computer knowledge and insights, their approaches to problem solving, their flexible and critical thinking and the values and excellence to be pursued and upheld; and
- employ a repertoire of strategies to provide students with multiple ways to acquire the knowledge, concepts and skills which the Curriculum encompasses.

(ii) Teachers are *knowledge facilitators*, for they

- are facilitators of students’ learning, not dispensers of facts and information in a student-centered teaching approach. The introduction of problem-based or project-based learning highlights the need for such a role.

(iii) Teachers are *resource consultants*, for they

- lead students to the art of self-learning by locating and securing tools, pools of resources and support to facilitate students’ learning at anywhere and anytime.

(iv) Teachers are *assessors*, for they

- assess individual student as well as the class as a whole by using multiple assessments and multiple dimensions of learning, both formatively and summatively;
- know how to move from assessment to decisions about teaching strategies and about where each student is in the continuum of learning in order to help him/her to improve, thus increasing the prospects for successful learning; and
- evaluate and reflect on their practice systematically and critically in light of student progress and pedagogical trends, and learn from experience.

(v) Teachers are *learners*, for they

- act as a role model for life-long learning through continuous self-updating and self-improvement in both computer and pedagogical knowledge and skills.

While each of the five teachers' roles has been described briefly and separately in the system of active learning, they are often interconnected and closely related. A teacher may indeed take on several roles simultaneously.

Teachers undoubtedly play a significant role in directing student learning in a dynamic flow of classroom activities. Assessment, on the other hand, helps students to understand curriculum expectations and can be a tool to improve their own learning. In fact, the kind of assessment used significantly influences what is learned and the degree of meaningful engagement by students in the learning process throughout the course.

V. Assessment

Purposes of Assessment

The overall purpose of student assessment in Computer Studies in a school setting is to gather information and make judgments about student quality of learning and achievement throughout the course. It is done for a variety of purposes:

- to provide qualitative information suitably to make fair judgment of student achievements;
- to enable students to gain information on what the Curriculum values, on promoting their self-knowledge about performance in order to facilitate and improve their learning;
- to evaluate and improve teaching effectiveness; and
- to incorporate into external examination.

Assessment for Learning

Traditional testing or assessment has always been done by the paper-and-pencil method. This kind of assessment, however, is inadequate in assessing students' performance and has the following short-comings:

- The demands made of 21st Century citizens as active learners who can articulate ideas, adapt and continuously reconstruct knowledge with critical thinking and problem solving abilities. These attributes cannot be adequately assessed by paper-and-pencil methods.

- The concept of learning has changed. Learning is not solely the acquisition of content knowledge. In fact, the Curriculum advocates the applications of computer knowledge and concepts, the development of communication and critical thinking skills, creativity and computer capabilities, and the fostering of positive values and attitudes towards the use of computer or information technologies. Paper-and-pencil tests, however, cannot measure the broad range of abilities fostered in the Curriculum. It is, indeed, only one way of collecting information about student learning.

- Paper-and-pencil testing can only engage students with certain kinds of learning styles such as verbal-linguistic and logical, leaving other dimensions of learning or ability out of the assessment process.

In the broader concept of assessment, it includes other assessment procedures such as observing student performances and critiquing student projects and reports. This broadens the kind of information that is collected about students and the way that this information is used in the facilitation of student learning.

Assessment can be used as a tool to facilitate student learning. Assessment for learning is the practice of collecting evidence of student learning so as to provide feedback to the learners and to the learning process, as well as to examine the progress of the learners so that more appropriate strategies for the next stage of development can be planned and implemented. It

- focuses on how and what students learn;
- integrates into learning and teaching;
- advocates a shared understanding of learning objectives and the criteria by which students are assessed;
- develops student's capacity for self-assessment and reflection;
- advocates the notion that students are given constructive guidance, feedback on work and opportunity to improve on their work; and
- recognises the full range of student achievements.

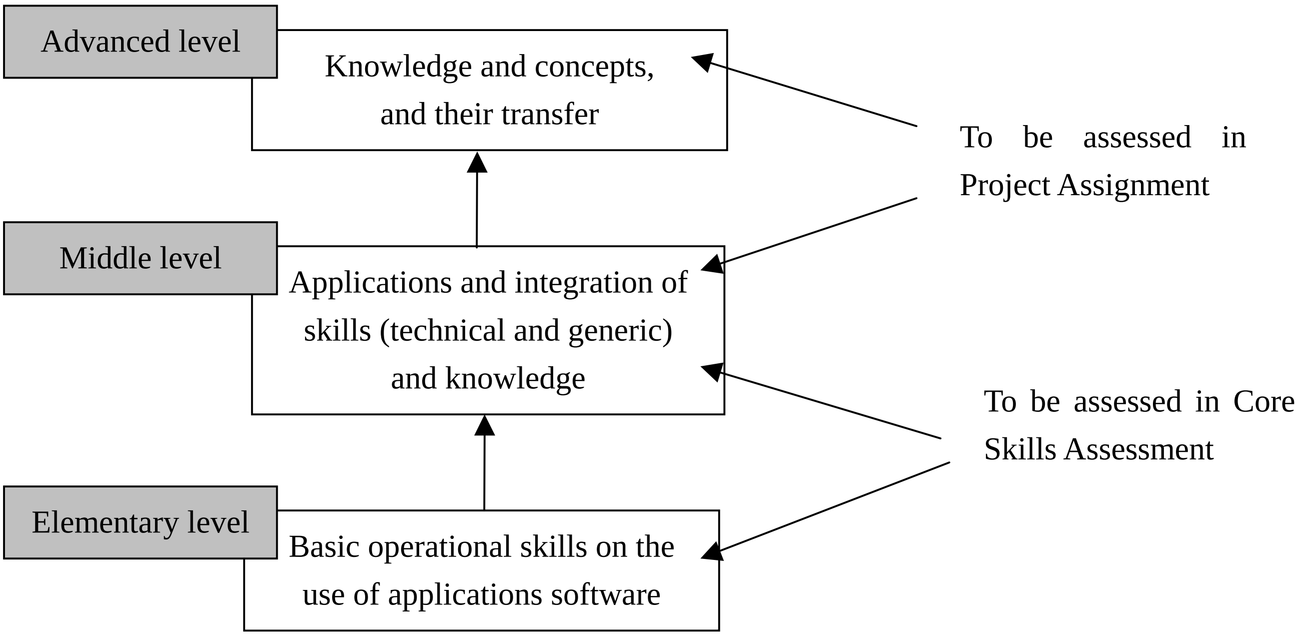
In assessment for learning, both the processes and the products of student work are equally important in student learning. Evaluation of student performance will be based on a broader concept of learning, ability and achievements. Teachers need to assess students' repertoire of learning strategies, skills in communicating with others, and knowledge and concepts as they are applied to real life and diverse contexts. This demands that teachers strive to provide new assessment techniques or procedures which are dynamic and continuous i.e. the School-based Assessment.

School-based Assessment (SBA)

In School-based Assessment teachers have the responsibility and flexibility for designing, constructing and administering assessment tasks, and for appraising student work. It recognizes teachers' professional judgments in the evaluation of student work. Based on the principles of assessment for learning, School-based Assessment should be on-going, formative, comprising multiple assessment and multidimensional with integrated / contextualised tasks.

Throughout the course of Computer Studies there are concepts and knowledge, applications of computer skills, generic skills (communication, creativity, problem solving and critical thinking) and project management skills, which students should master or develop. There should, however, be a balance between the knowledge, concepts and understanding outcomes, and skills (technical and generic) outcomes.

School-based Assessment is structured around two assessment components, comprising the Core Skills Assessment and a Project Assignment. These are used to measure the three cognitive levels prevailing in the course, as depicted in the following diagram


Advanced level
Knowledge and concepts, and their transfer
To be assessed in Project Assignment
Middle level
Applications and integration of skills (technical and generic) and knowledge
To be assessed in Core Skills Assessment
Elementary level
Basic operational skills on the use of applications software

School-based Assessment of the three cognitive levels in A-level Computer Studies

Core Skills Assessment

One of the assessment objectives involves practical skills that cannot be readily and adequately assessed by written examinations. The Core Skills Assessment aims at evaluating whether students can perform the required tasks, irrespective of how long it may take them to grasp the skills. In Core Skills Assessment, students are required to demonstrate their competence through various problem-solving tasks using major features of computer applications software and a wide range of abilities and skills stipulated in the Curriculum. The Core Skills Assessment is an ongoing process for the development of students' practical skills. Not only should teachers assess students' performance in Core Skills Assessment, but also analyse students' performance so as to provide formative feedback to facilitate student learning. Teachers can help students to develop a portfolio of learning and assessment throughout the course and use feedback to inform students of their strengths and weaknesses in order to motivate and reinforce their learning.

The assessment on students' core skills, however, should not be confined within class time nor solely within the context of this Curriculum, for computer skills have now become everyday life skills. Students may demonstrate their competency in computer skills in an interdisciplinary manner.

Project Assignment

The Project Assignment is a powerful instrument to assess nearly all dimensions of learning; knowledge and concepts, technical and generic skills, and values and attitudes. It taps the power and diversity of active learning of students. In doing the Project Assignment, students should be asked to use a wide range of cognitive processes and abilities such as to perform, create, produce or do something that requires them to use higher-level problem-solving skills or thinking skills to analyze and to interpret their work and to reflect or evaluate their quality of learning. What is assessed is what is valued. So in designing the assessment criteria, teachers should include not just the knowledge aspect but also other learning dimensions or values that the Curriculum advocates.

Public Examination

The public examination consists of two written papers, Paper 1 and Paper 2, each carries 40% of the overall subject mark, and a School-based Assessment component (Paper 3) occupying 20% of the overall subject mark. In Paper 1 and Paper 2, there will be two sections, A and B. Section A will consist of short questions while Section B will comprise of long questions. Paper 1 will be a common paper to Advanced Supplementary Level Computer Applications. Paper 3 is a School-based Assessment component in which students will be assessed internally at schools by their teachers. As mentioned previously, it will consist of the Core Skill Assessment and a Project Assignment.

The detailed assessment criteria, rules and regulations as well as assessment modes can be found in the Hong Kong Advanced Level Examination Regulations and the Advanced Supplementary Level Computer Applications School-Based Assessment Handbook published by the Hong Kong Examinations and Assessment Authority. Teachers and students should refer to these documents for further information.

VI. References

Reference Books for Office and Internet Applications

Title	Author	Publisher	Year of Publication
A Guide to Microsoft Office 2000 Professional	Bruce Presley, Beth Brown, Elaine Malfas and Vickie Grassman	Lawrenceville	2000
Computer Networks and Internets with Internet Applications (4th Edition)	Douglas Comer, Ralph Droms	Prentice Hall Press	2003
How the Internet Works	Preston Gralla	Que	2000
Internet Search Techniques (3rd Edition)	Karin Rex	Element K	2001
Computer Virus 百毒防治	陸金山、王岩增	知城	2002
Excel 2003 函數與統計應用實務	吳權威、呂琳琳	網奕	2003

Reference Books and Websites for Databases

Title	Author	Publisher	Year of Publication
An Introduction to Database Systems (8th Edition)	C. J. Date	Addison-Wesley	2003
A Visual Introduction to SQL (2nd Edition)	David Chappell, J. Harvey Trimble	Wiley	2001
Database Demystified	Andy Oppel	McGraw-Hill / Osborne	2004
Database Processing (9th Edition)	David Kroenke	Prentice Hall Press	2003
Database Systems: Design, Implementation and Management (5th Edition)	Rob and Coronel	Course Technology	2002

Title	Author	Publisher	Year of Publication
Fundamentals of Database Systems (4th Edition)	Ramez Elmasri, Shamkant Navanthe	Addison Wesley	2004
MySQL: Essential Skills (1st Edition)	John Horn, Michael Grey, LLC Interstate Software	McGraw-Hill Osborne Media	2004
MySQL (2nd Edition)	Paul DuBois	Sams	2003
PHP and MySQL for Dynamic Web Sites (1st Edition)	Larry Ullman	Peachpit Press	2003
SQL: A Beginner's Guide (2nd Edition)	Robert Sheldon	McGraw Hill / Osborne	2003

Websites

Database Normalization Basics	http://databases.about.com/od/specificproducts/a/normalization.htm
Entity-Relationship Modeling	http://www.iit.edu/~abdemet/ERModeling.htm
Introduction to SQL	http://www.users.nac.net/dmdgm2/sqlsubquery.html#G301
Introduction to the relational model	http://www.cs.ust.hk/~wilfred/COMP630H/2Relational.pdf
Introduction to Structured Query Language	http://www.intermedia.net/support/sql/sqltut.asp
Introduction to Structured Query Language	http://riki-lb1.vet.ohio-state.edu/mqlin/computec/tutorials/SQLTutorial.htm
Normalization	http://www.educationandskills.com/database/normalisation/
Normalization and Database Tables	http://personal.cityu.edu.hk/~ismikeho/dm2/dmlec4.htm
Problem of Data Redundancy	http://www.cs.ust.hk/~wilfred/Bobby/public_html/fd/fd_redundancy.html
Relational Model Concepts	http://www.ieem.ust.hk/dfaculty/ajay/courses/ieem230/lecs/rd-des/rd-des.html
SQL Tutorial	http://www.w3schools.com/sql/default.asp

Reference Books and Websites for SOHO Networking

Title	Author	Publisher	Year of Publication
Computer Networks and Internets (4th Edition)	Douglas E. Comer	Prentice Hall	2003
Home Networking with Microsoft Windows XP Step by Step	Heather T. Brown, Matthew Danda	Microsoft Press	2001
Home Office Computing Survival Guide	TechRepublic	CNET	2002
Local Area Network Management, Design & Security	Arne Mikalsen, Per Borgesen	John Wiley & Sons	2002
Network Architecture & Design "A Field Guide for IT Professionals"	Dimarzio, J.F. DiMarzio	Sams	2001
Poor Richard's Home and Small Office Networking: Room-to-Room or Around the World	John Paul Mueller	Top Floor Pub	2001
Small Business Solutions for Networking	Alan Neibauer	Microsoft Press	2000
SOHO Networking: A Guide to Installing a Small-Office/Home-Office Network	Pete Moulton, Peter Moulton	Prentice Hall PTR	2002
PCDIY 架網路 快、易、通	施威銘研究室	旗標出版股份有限公司	2003
PCDIY 2005 無線網路選購、架設、活用	施威銘研究室	旗標出版股份有限公司	2005
PCDIY 2004 網路自己裝	施威銘研究室	旗標出版股份有限公司	2004
無線網路架設實務	施威銘研究室	旗標出版股份有限公司	2002
802.11 無線區域網路理論與實務	顏春煌	旗標出版股份有限公司	2004
無線寬頻上網大作戰	施威銘研究室	旗標出版股份有限公司	2002

Title	Author	Publisher	Year of Publication
E-Commerce 網站防駭實務	葉育如	旗標出版股份有限公司	2002
防毒防駭全攻略	程秉輝	旗標出版股份有限公司	2004
資訊與網路安全技術	粘添壽、吳順裕	旗標出版股份有限公司	2004
數位達人系列－無線網路基礎概念圖解即效養成	陳人豪	松崗電腦圖書資料股份有限公司	2004
數位達人系列－網路基礎概念圖解即效養成	葉雅美	松崗電腦圖書資料股份有限公司	2004
輕鬆設定家庭網路	傑洛	碁峰資訊股份有限公司	2004
網路安全徹底研究	Robert Bragg、Mark Rhodes-Ousley、Keith Strassberg	學貫行銷股份有限公司	2004
2004 區域網路與寬頻實務	王緒溢	碁峰資訊股份有限公司	2004
網路疑難急救	馬文剛 ，吳宏	萬里機構‧萬里書店	2004
電腦 DIY 2004－區域網路架設秘笈	PC Man	文魁資訊股份有限公司	2004
圖解寬頻網路架設、檔案分享徹底活用	亦向工作室	上奇科技股份有限公司	2003
區域網路架設與實務應用	Alan Neibauer	文魁資訊股份有限公司	2002

Websites

Client/Server Vs Peer to Peer Networking, *Internet Xtreme*	http://www.xtreme.net.au/Network/server.asp
Download these SOHO networking checklists, *TechRepublic*	http://techrepublic.com.com/5100-6265-5054267.html
End-user support, *The Office of Government Commerce*	http://www.ogc.gov.uk/sdtoolkit/reference/deliverylifecycle/end_user_supp.html
Home computer networking and Internet connection sharing, *HomeNetHelp.com*	http://www.homenethelp.com/
Home Networking, *Putergeek.com*	http://www.putergeek.com/home_network/
Home Network Setup - Networking Advisor, *About Inc.*	http://compnetworking.about.com/od/homenetworking/l/blhomeadvisor.htm
How to Build a Safe, Secure Network, *PCWORLD*	http://www.pcworld.com/howto/article/0,aid,115066,pg,3,00.asp
How to Build a Wireless Home Network - Tutorial, *About Inc.*	http://compnetworking.about.com/cs/wirelessproducts/a/howtobuildwlan.htm
Internet Connection Sharing Linux as a broadband router, *HomeNetHelp.com*	http://www.homenethelp.com/web/howto/linux-share-internet.asp
Linux Home Networking, *www.linuxhomenetworking.com*	http://www.chinalinuxpub.com/doc/www.siliconvalleyccie.com/
Reinventing the Workplace: An Introduction to Telecommuting, *Meridian Technology Corporation*	http://www.officewizard.com/books/Telecommuting/02785_Intro_Telecommuting.pdf
The History and Direction of Telecommuting, *Stallion Technologies*	http://www.stallion.com/html/solutions/telecommuting-directions.html
Step-By-Step: Set Up a Network in a Snap, *PCWORLD*	http://www.pcworld.com/howto/article/0,aid,68763,pg,2,00.asp
Step-By-Step: Share Your Net Connection, *PCWORLD*	http://www.pcworld.com/howto/article/0,aid,108740,00.asp
Telecommuting Requirements Checklist, *Government of Alberta*	http://www.pao.gov.ab.ca/health/flexible-work/telecommuting/appendix-2.htm

Telecommuter Handbook, *The TMA Group*	http://www.tmagroup.org/TelGuide.html
Top 7 Tips for Wireless Home Network Security, *About Inc.*	http://compnetworking.about.com/od/wirelesssecurity/tp/wifisecurity.htm
WEP - Wired Equivalent Privacy, *About Inc.*	http://compnetworking.about.com/cs/wirelesssecurity/g/bldef_wep.htm
Wireless LAN Security FAQ, *Internet Security Systems*	http://www.iss.net/wireless/WLAN_FAQ.php#[2]%20What%20are%20the%20major%20security%20risks%20to%20802.11b?
Wireless LAN Step by Step Guide, Computing Services, *The University of Edinburgh*	http://www.ucs.ed.ac.uk/nsd/access/wsteplist.html
Wireless Networking Mini-Tutorial, *WKMN Training*	http://www.wkmn.com/newsite/wireless.html
VPN Router with IPSec Client software - tutorial, *HomeNetHelp.com*	http://www.homenethelp.com/vpn/router-config.asp
VPN Tutorial - An introduction to VPN software, VPN hardware and protocol solutions, *About Inc.*	http://compnetworking.about.com/od/vpn/l/aa010701a.htm
VPN - Virtual private Networking, an overview, *HomeNetHelp.com*	http://www.homenethelp.com/vpn/

Reference Books for Computer Organization

Title	**Author**	**Publisher**	**Year of Publication**
Computer Systems Design and Architecture (2nd Edition)	Vincent P. Heuring, Harry F. Jordan	Prentice Hall	2004
Computer Systems	Bryant, Randal E., O’Hallaron, David R.	Prentice Hall	2002
Computer Systems Architecture	Rob Williams	Addison Wesley	2001
Reliability of Computer Systems and Networks	Shooman, Martin L.	John Wiley and Sons	2002

Reference Books for Systems Development

Title	Author	Publisher	Year of Publication
"A" Level Information Technology	P. M. Heathcote	Payne-Gallway	1998
Management Information Systems (6th Edition)	Kenneth C. Laudon and Jane P. Laudon	Prentice Hall	2000
Managing Information Technology (4th Edition)	E. Martin, C. Brown, D. WeHayes, J. Hoffer and W. Perkins	Prentice Hall	2001

Reference Books and Websites for Programming

Title	Author	Publisher	Year of Publication
An Introduction to Programming using Visual Basic 6.0 4/E Updated	David I. Schneider	Prentice Hall	2004
C: A Reference Manual (5th Edition)	Samuel P. Harbison	Prentice Hall	2002
C Programming FAQs: Frequently Asked Questions	Steve Summit, Deborah Lafferty	Addison Wesley	1995
Computer Algorithms: Introduction to Design and Analysis (3rd Edition)	Sara Baase, Allen Van Gelder	Addison Wesley	1999
Java: How to Program (6th Edition)	Paul J. Deitel	Pearson	2005
Learning to Use Visual Basic for Standard Grade Computing (with CD -ROM)	A. Holmes	Heinemann	2003

Title	**Author**	**Publisher**	**Year of Publication**
Schaum's Outlines: Programming with Java	John. R Hubbard	McGraw Hill	2004
Schaum's Outlines: Programming with Pascal	Byron Gottfried	McGraw Hill	1994
Turbo Pascal	Koffman Elliot	Addison Wesley	1998
C 語言入門（附 CD）	林邦傑	知城	2002
C 語言程序設計基礎與應用	孫振業編寫	海洋	2004
Java2 教學範本	古昌弘	金禾資訊	2005
PASCAL 程序設計（第二版）	鄭啓華	清華大學	1996
Turbo Pascal 入門與應用徹底剖析	洪錦魁	松崗	1998
Visual Basic 6.0 程式教學手冊（附光碟）	志佳工作室	博碩文化	2004
深入淺出 Java 語言程序設計（附光盤）	李雲山、呂傑武	中國青年	2003
精彩 Visual Basic 6.0 學習範本	吳逸賢、吳目誠	網奕資訊科技	2004

Websites

C Programing; University of Strathclyde Computer Centre	http://www.its.strath.ac.uk/courses/c/
Free Programming and Computer Science Books; Google Group	http://www.techbooksforfree.com/
Howstuffworks “How C Programming Works”; HowStuffWorks, Inc.	http://www.howstuffworks.com/c.htm
Informit Safari Tech Books Online; Pearson Education, InformIT	http://safari.informit.com/
Java For Students; Ben Golding	http://www.javaforstudents.co.uk/
Pascal Programming; Brian Brown/Peter Henry/CIT	http://www.geocities.com/SiliconValley/Horizon/5444/pstart.htm
Programmingtutorials.com	http://www.programmingtutorials.com/Default.aspx
Taoyue.com: Learn Pascal tutorial; TaoYue.com	http://www.taoyue.com/tutorials/pascal/contents.htm
The Java Tutorial; Sun Microsystems, Inc.	http://java.sun.com/docs/books/tutorial/
Visual Basic Examples; Saul Greenberg	http://pages.cpsc.ucalgary.ca/~saul/vb_examples/#Tutorials