

**Information and Communication Technology Curriculum and Assessment Guide
(S4-6)
Supplementary Notes (Clean Version)**

Curriculum Framework

Revised Curriculum Framework (2025 HKDSE onward)

The Compulsory Part (144 hours)		
A. Information Processing (37 hours)	B. Computer System Fundamentals (20 hours)	
C. Internet and its Applications (31 hours)	D. Computational Thinking and Programming (48 hours)	
E. Social Implications (8 hours)		
+		
The Elective Part (76 hours) (Choose any two)		
A. Databases	B. Web Application Development	C. Algorithm and Programming
+		
School-based Assessment (30 hours)		

The Compulsory Part

The Compulsory Part comprises five modules:

A. *Information Processing*

Introduction

This module provides students with a fundamental understanding of what information systems and information processing entail, and how data are represented inside a computer. Students will learn that different software is used to process different types of data. They will be given practical tasks to process and present information. Built on the basic knowledge, understanding and skills of spreadsheet and database acquired by students in junior secondary. This will enhance students' personal productivity in their work or study, and enable them to apply their learning effectively to problem-solving and data analysis.

Learning Objectives

Students will learn about:

- information systems and processes in real-life contexts;
- the difference between information and data;
- how data are organised and represented inside a computer;
- manipulate and analyse data; and
- how advances in ICT foster the development of the Information Age and its impact on society.

The time allocation for the module is about 37 hours.

Details

This module comprises four topics: "Introduction to Information Processing", "Organisation and Data Control", "Data Representation" and "Data Manipulation and Analysis" Details of the four topics are summarised below:

Topic	Learning Outcomes	Remarks
<p>a. Introduction to Information Processing (3 hours)</p>	<ul style="list-style-type: none"> • Identify and examine the components of an information system. • Distinguish between various information processes. • Realise the difference between data and information, and identify different types of data as image, audio, video and text. • Define Information Age and discuss the importance of information literacy in a knowledge-based society. 	<p>The components include the purposes, data, processes, technologies and personnel.</p> <p>Information processes include data collection, organisation, analysis, storage, processing, transmission and presentation. Examples from daily life, both computer and non-computer-based processes, should be used to consolidate and enhance students' understanding of the activities involved.</p> <p>How the development of technologies leads to the emergence of the Information Age, and how information can be flexibly and analytically converted into knowledge in modern society should be discussed.</p>

Topic	Learning Outcomes	Remarks
b. Data Organisation and Data Control (4 hours)	<ul style="list-style-type: none"> • Identify data, records, fields, files and databases in the hierarchical organisation of data. • Explain how records can be organised, stored and retrieved. State the advantages, disadvantages and applications of the two types of file access: direct access and sequential access. • Discuss the needs of data control. • Describe how errors can be detected by using validation and parity checking, and prevented by verification and validation. 	
c. Data Representation (10 hours)	<ul style="list-style-type: none"> • Distinguish between analog and digital data. State applications or situations where conversion of analog to digital data is required, or vice versa. • Explain why IT uses digital data 	<p>The relationship between the number of bits and number of patterns/combinations available should be mentioned (e.g. three bits can be used to represent eight colours).</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Convert integers from denary numbers to binary numbers or hexadecimal numbers, or vice versa. • Perform simple calculations (addition and subtraction only) on binary numbers and analyse overflow errors. • Know how characters are represented by using common international standards such as American Standard Code for Information Interchange (ASCII), the Big-5 code, the Guobiao (GB) code and the Unicode. • Know briefly how different multimedia elements are digitised. Convert them into different file formats and compare them for storing the same data. 	<p>Adopt two's complement for the representation of negative integers.</p> <p>In understanding errors, minimum and maximum numbers an n-bit can hold (maximum 2 bytes) should be considered.</p> <p>The relationship between the size of the character set and the representation should be explained. Recall of specific codes is not required.</p> <p>Common file formats such as bmp, png, jpg, wav, mp3, avi, mpeg4, txt, docx, odt and pdf.</p>
<p>d. Data Manipulation and Analysis (20 hours)</p>	<ul style="list-style-type: none"> • Describe and use basic features of spreadsheets to solve problems. • Demonstrate data manipulation techniques in spreadsheets. 	<p>The use of cell references in formulas, and functions, together with mathematical, logical and relational operators should form part of the content.</p> <p>This includes filtering, searching and sorting data using single or multiple criteria. The manipulation of data dynamically in multiple worksheets should also be introduced.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Apply spreadsheets as a data analysis tool by using a pivot table (and pivot chart), and “what-if” scenarios. • Apply the concepts of data organisation to create and maintain a simple database using a Database Management Systems (DBMS) tool. • Create and use a form for data entry. • Practise data extraction and manipulation by querying a database and create reports. 	<p>Through task-based activities, apart from the basic features and functions (sum, sub-total and average) of a pivot table, students should be led to observe and analyse the interdependency of data by varying the fields in a pivot table.</p> <p>Together with charting, simulating real-life situations and “what-if” scenarios, students should learn to identify trends, to make informed judgments, and to produce meaningful predictions which are required as critical thinkers throughout their careers.</p> <p>This includes the selection, filtering and sorting of data using query in a single database table. Students should be able to trace and interpret simple Structured Query Language (SQL) statements. Though sophisticated reports are not required, they should create and format reports for intended users / audience.</p>

B. Computer System Fundamentals

Introduction

This module provides students with a basic understanding of how the different components of a computer system work together to perform computational tasks. This includes the learning of the functional units of a computer, the system software, and the different modes of operation for different applications.

Learning Objectives

Students will learn about:

- the functions and properties of the major components of a computer system and how these components interact together to perform tasks;
- the functions and properties of major peripheral devices, and their uses in specific situations;
- the use of different utility programs in managing systems and files;
- the capabilities of different operating systems, and the basic concepts of a computer network and its applications; and
- different modes of operation.

The time allocation for the module is about 20 hours.

Details

This module comprises two topics: “Basic Machine Organisation” and “System Software”. Details of the two topics are summarised below:

Topic	Learning Outcomes	Remarks
<p>a. Basic Machine Organisation (14 hours)</p>	<ul style="list-style-type: none"> • Explain the functions of hardware within a computer system, namely input and output devices, processing units, bus system and storage devices (both primary and secondary). • Explain the structure and functions of a CPU and its components. • Outline the steps in the fetch-decode-execute cycle and store using a single processor, and describe the roles of and the interdependence among components, registers and buses in the machine cycle. 	<p>Processing units include central processing unit (CPU) and graphics processing unit (GPU).</p> <p>Students should know how CPU is measured in terms of frequency. Units such as microsecond, nanosecond and picosecond should also be introduced.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> <li data-bbox="817 263 1420 518">• Describe the functions and characteristics of Random Access Memory (RAM), Read Only Memory (ROM) and memory cache. Realise the relationship among the size of the memory, the memory address, word length and the performance of the computer. <li data-bbox="817 742 1420 965">• Describe the features, advantages, disadvantages and applications of the input and output devices. Select and justify the use of appropriate devices for collecting and displaying information in a given context. <li data-bbox="817 1005 1420 1157">• Describe the functional characteristics of storage devices in terms of random or sequential access, volatile or non-volatile, data transfer rate and storage capacity. 	<p data-bbox="1438 263 2036 406">Students' understanding of main memory should be extended to the current version / technologies in RAM and ROM in terms of capacity and data access rate.</p> <p data-bbox="1438 446 2038 702">The meanings of units such as terabytes, gigabytes, megabytes and kilobytes should be introduced. Also, the distinction between prefixes used in computer and the Syst�me International (SI) notation should also be made (e.g. 1KB = 1024 Bytes, not 1000 Bytes).</p> <p data-bbox="1438 742 2038 925">Students should know the various hardware devices used for collecting and displaying different types of data such as image, audio, video and text. The technical details on how each device operates are not required.</p> <p data-bbox="1438 1005 2038 1260">Examples of storage devices are magnetic disk, optical disk, flash memory, magnetic tape and network storage. In understanding the characteristics of storage devices, students should be made aware of the trend to faster and greater storage capacity but smaller physical size over time.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> Outline the latest developments in computer systems including processor capabilities, primary memory technologies, secondary storage devices and data communications. 	<p>Technical details are not required.</p>
<p>b. System Software (6 hours)</p>	<ul style="list-style-type: none"> Know the functions of system software and applications software, and the relationship between hardware, system software, applications software and users. Outline the basic functions of an operating system and describe some common operating systems, and their differences and applications. State the functions and needs of utility programs and driver programs. Distinguish the characteristics and applications of various modes of operation. 	<p>Examples of the utilities are data compressors, virus checkers, file managers, defragmentation software and system monitoring software. Technical details of these programs are not required but demonstration or practice on their use should be given.</p> <p>Modes of operation to be considered are batch processing, real-time processing, parallel processing, distributed processing, and virtualisation.</p>

C. Internet and its Applications

Introduction

This module is designed to acquaint students with Internet fundamentals. It encompasses the concepts of Internet access, services and applications of the Internet, and elementary web page design. Students will also be given the opportunity to critically analyse the reliability of the information retrieved from the Web and appreciate the impact of the Internet on various activities in society.

Learning Objectives

Students will learn about:

- how to connect to the Internet, and the hardware, software and Internet Service Provider (ISP) involved in accessing the Internet;
- the personal, social and commercial activities that are available on the Internet;
- how to participate in various Internet activities such as searching for information, sharing opinions, and exchanging messages and files;
- the technologies involved in transmitting and displaying multimedia elements on the Internet;
- the design and construction of simple web pages for an intended audience;
- the potential threats on the Internet and measures to reduce them; and
- the need to use ICT safely, sensibly, legally and ethically.

The time allocation for the module is about 31 hours.

Details

This module comprises four topics: “The Networking and Internet Basics”, “Internet Services and Applications”, “Elementary Web Authoring” and “Threats and Security on the Internet”. Details of the four topics are summarised below:

Topic	Learning Outcomes	Remarks
<p>a. Networking and Internet Basics (9 hours)</p>	<ul style="list-style-type: none"> • Define and compare Local Area Network (LAN) and Wide Area Network (WAN). • Know the formats and functions of IPv4 and IPv6. • Discuss the common services available in a networked environment. • Explain the functions of the hardware required for a network. • Compare common methods for Internet access in terms of speed, cost, security and availability. • Understand the need for communications software and communication protocols. 	<p>Technical details are not required.</p> <p>Services include internal communications, conferencing and resources sharing.</p> <p>This includes communication links (fibre optics, microwave, Unshielded Twisted Pair (UTP) cable, satellite, etc.), modem, network interface card, network connecting devices (switch and router, etc.).</p> <p>The common industry standards for wireless computer networks should be introduced to students. Relevant concepts including frequency, bandwidth, interference and roaming, etc. are required.</p> <p>This includes connections in wireless and broadband etc.</p> <p>This includes simple concepts of TCP/IP.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Describe how data is transmitted over the Internet and understand concepts of Uniform Resource Locator (URL), Domain Name System (DNS), Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS). 	
b. Internet Services and Applications (5 hours)	<ul style="list-style-type: none"> • Formulate an effective strategy for searching for specific information on the Web by using search-engines, and critically analyse the sources of information. • Identify various graphics, audio, video file formats suitable for web pages. Use plug-ins and players for the multimedia elements found on the Internet. • Apply various services such as file transfer, remote logon, online chat, discussion forum and email on the Internet. • Describe the concepts of streaming technology and its applications in voice mail, videoconferencing, and webcasting etc. on the Internet. 	<p>Technical details of streaming technology are not required.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> Value the significance of the development and expansion of the Internet for various activities in society. 	<p>Appreciation of the use of Internet applications for the improvement of human life should be fostered among students. For instance, smart city can be achieved with Internet of things (IoT) and cloud services.</p>
<p>c. Elementary Web Authoring (3 hours)</p>	<ul style="list-style-type: none"> Recognise the basic constructs of Hypertext Markup Language (HTML) which is a means to address cross-platform issues. Discuss the organisation of web pages for an intended audience and upload them onto the World Wide Web. 	<p>The organisation of information includes ease of navigation, appropriate placement of links, tables, frames and multimedia elements, colour combinations, background design, font size and style, for an intended audience.</p> <p>Students are not required to memorise HTML codes.</p>
<p>d. Threats and Security on the Internet (14 hours)</p>	<ul style="list-style-type: none"> Describe the potential risks caused by the common network security threats. 	<p>The threats include virus, worm and Trojan programs, spyware, ransomware, unauthorised access, interception, intrusion via dynamic web pages and Denial of Service (DoS) attack, etc.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Explain authentication and authorisation as a means to control access of information on the Internet. • Know about security used in electronic transactions. • Be aware of the latest developments in security measures. 	<p>Basic concepts include the authentication methods for individuals, types of tokens used in authentication processes, and the procedures of authenticating a digitally signed document by obtaining a digital certificate of the signed body.</p> <p>The concepts of Secure Sockets Layer (SSL) in secured transmission in e-commerce should be introduced.</p> <p>Other security measures in online transaction such as smart cards, security tokens, digital certificates and mobile Short Message Service (SMS) should also be introduced.</p>

D. Computational Thinking and Programming

Introduction

Computational Thinking (CT) is an approach to solving problems in a way that can be implemented with a computer. Students become tool builders instead of tool users through a set of concepts such as abstraction, algorithm and automation. This module is designed to teach students how to go through a systematic process of thinking when solving problem (abstraction), formulating steps for solutions (algorithmic thinking) and writing computer program (programming/coding) to produce the solution. Advanced programming concepts are introduced in *Algorithm and Programming*.

Learning Objectives

Students will learn about:

- the systematic approach to problem-solving;
- the application of concepts of systematic problem-solving to real-life problems;
- the use of pseudocode and/or a program flowchart to represent the algorithm;
- creative design and development of computer program;
- how to illustrate different programming paradigms with appropriate programming languages;
- how to identify the objectives of an algorithm, trace the logical flow and examine values of variables during execution; and
- various ways of solving the same problem, and the differences between them.

The time allocation for the module is about 48 hours.

Details

This module comprises four topics: “Problem-Formulation and Analysis”, “Algorithm Design”, “Program Development” and “Program Testing and Debugging”. Details of the four topics are summarised below:

Topic	Learning Outcomes	Remarks
a. Problem-Formulation and Analysis (5 hours)	<ul style="list-style-type: none"> • Define a problem and its scope. • Analyse a problem by identifying required inputs and outputs as well as stating the processes required. • Solve a problem by decomposing them into smaller and manageable sub-problem. • Identify common elements across similar problems. 	<p>Examples:</p> <ul style="list-style-type: none"> • Calculate interest on mortgages and list the instalments • Find the Body Mass Index (BMI) to monitor for healthy weight • Program a robot to detect and trace lines <p>The sub-problems, for instance, may represent the input, process and output of the solution to the problem.</p> <p>Example:</p> <ul style="list-style-type: none"> • Identify the patterns of methods for sorting the height of a group of students in ascending order and then modify the methods to sort the weight of a group of students in descending order. • Identify the patterns of methods for programming a robot to move in a square and then modify the programs to let the robot move in other polygons.
b. Algorithm Design (12 hours)	<ul style="list-style-type: none"> • Perform a dry run of a set of steps to determine its purpose and/or output. 	

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Define algorithm. Use pseudocode and program flowchart as methods for representing algorithms. • Outline and discuss the input and output requirements of a problem, and design an appropriate user interface. • Recognise the uses and nature of simple data types and data structures in solving a problem. • Select appropriate data types for the solution to a particular problem and discuss the merit of the chosen types. • Design and construct standard algorithms involving basic control structures. • Create and examine algorithms such as to load and print an array, and to add or delete an item from an array. • Produce trace table to show value of variables at each stage in set of steps. 	<p>Simple data types are restricted to integer, real, character and Boolean while simple data structures are limited to string and one-dimensional array. Boolean logic (AND, OR, NOT) and truth tables should be introduced.</p> <p>The control structures are sequence, selection (binary and multi-way) and iteration (pre-test, post-test and for loops).</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Locate logic error in an algorithm, and correct or modify an algorithm to remove the errors or for changes in task specification. • Describe the advantages of modularity in designing computer solutions. 	<p>When designing a solution to a complex problem, students should be encouraged habitually to use the modular approach to structure the algorithm.</p>
<p>c. Program Development (20 hours)</p>	<ul style="list-style-type: none"> • Understand and use variables, constants, and simple lists (1-D array) in different problem contexts. • Use operators, expressions, assignment statements, input and output statements. 	<p>Examples of arithmetic operators include addition, subtraction, multiplication, division, and modulus.</p> <p>Examples of relational operators include equal to, not equal to, greater than, greater than or equal to, less than and less than or equal to.</p> <p>Examples of Boolean expression include AND, OR and NOT.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Understand and use sequence, selection and iteration (nested loop is not required) constructs to create a program • Produce programming solution for a given problem. 	<p>Examples:</p> <ul style="list-style-type: none"> • Find the minimum, maximum and average values in a list • Search for an item in a list and report the result of the search • Find the length of a string of characters • Extract required characters from a string of characters • Count the number of items, which meet specified criteria in a list • Check if the values in a list are in order • Use of mathematical formulas
<p>d. Program Testing and Debugging (11 hours)</p>	<ul style="list-style-type: none"> • Apply data validation check to design appropriate test data. • Understand and describe types of program errors: syntax, logic and run-time; explain why they occur and debug. • Compare different solutions to the same problem. 	<p>Students need to identify boundary cases.</p> <p>Comparison of the steps of operation and resource usage of different programs to solve the same problem should be encouraged.</p>

E. Social Implications

Introduction

This module provides students with an understanding, and ethical analysis, of various issues arising from the use of ICT. These issues have economic, legal, social, ethical and security consequences. Students should be given experience in discussing and debating these issues to develop the analytical and interpretive skills required to construct their own normative practices in the use of ICT. This module encourages students to act in a socially responsible, ethical and legal way in using the technologies throughout their studies, careers and other areas of their lives.

Although these issues are introduced specifically in this module, they should also be re-considered and re-addressed in various other parts of the course as a means to strengthen students' awareness of them.

Learning Objectives

Students will learn about:

- basic concepts of technological innovations;
- equity issues relating to access to ICT;
- ethical considerations on the use of ICT;
- health hazards and preventive measures in using ICT; and
- major issues regarding intellectual property and privacy.

The time allocation for the module is about 8 hours.

Details

This module comprises three topics: “Technological Innovations”, “Health and Ethical Issues” and “Intellectual Property”. Details of the three topics are summarised below:

Topic	Learning Outcomes	Remarks
a. Technological Innovations (3 hours)	<ul style="list-style-type: none"> • Understand the basic concepts of technological innovations and its applications. 	<p>Examples include pattern recognition through artificial intelligence (AI) and data science, 3D printing technologies, augmented reality (AR) and virtual reality (VR).</p> <p>Students should have practical experience with these technologies.</p>
b. Health and Ethical Issues (3 hours)	<ul style="list-style-type: none"> • Identify health hazards associated with the use of ICT, and propose good ergonomic practices when using them. • Realise the importance of equity of access. • Discuss the ethical considerations on the use of ICT. 	<p>In ergonomics, consideration includes the reduction of injuries which arise from repetitive strain injury (RSI), the suitable design and placement of the furniture, the design of the software (with user-friendliness features such as ease of use and a consistent user interface) and work environment.</p> <p>Students should state the pros and cons of freedom of information on the Internet. They should also know equity issues in terms of the digital divide, gender equity and access for the disabled from local and global perspectives.</p>

Topic	Learning Outcomes	Remarks
<p>c. Intellectual Property (2 hours)</p>	<ul style="list-style-type: none"> • Understand the basic ideas of intellectual property and copyright. • Understand the benefits and risks of different licensing schemes such as freeware, shareware, open source software and copyrighted software from the perspectives of users and software developers. • Relate acts of possible infringement of copyright in software and Internet piracy. • Be aware of ways to reduce intellectual property theft on digital property. • Know some of the legal consequences, especially in education, related to the infringement of copyright in Hong Kong. 	<p>Students should discuss the social, legal and economic implications of acts of infringement of copyright. They should also develop the habit of acknowledging the source of information and be aware of the appropriate use of multimedia materials, and the possible crimes and consequences of illegal uses and broadcasting of these materials.</p> <p>Examples of measures are the use of a digital watermark and digital signature.</p> <p>Teachers can quote some of the legal consequences related to infringement of copyright.</p>

The Elective Part

The Elective Part comprises three options.

A. Databases

Introduction

This option is designed to provide students with the fundamental concepts of databases and relational database design. Students will learn how to construct simple data models using Entity Relationship (ER) diagrams and to appreciate the importance of good database design. They will also learn to use SQL to construct, manipulate and retrieve information from a relational database. In addition, students will be exposed to database security, integrity, and data privacy issues. Through studying this option, students will acquire a basic understanding of the concepts, skills and applications of databases, and elementary data modelling concepts.

Students should have acquired the necessary knowledge and skills in the Compulsory Part – module A *Information Processing* before progressing to this option.

The sub-topics under topic (a) “Relational Databases Concepts” in the present option are designed as extensions of topic (b) “Data Organisation and Data Control” in the Compulsory Part – module A *Information Processing*.

Learning Objectives

Students will learn about:

- concepts and applications related to databases and the DBMS;
- the basic concepts of a relational database, and the construction, manipulation and extraction of information from a relational database using SQL;
- the construction of simple data models using the ER diagrams methodology;
- the importance of good database design as a blueprint for the development of a database management system; and
- the importance of data privacy and measures for its improvement.

The time allocation for the option is about 38 hours.

Details









The option *Databases* comprises three topics: “Relational Databases Concepts”, “SQL”, and “Database Design Methodology”. Details of the three topics are summarised below:

Topic	Learning Outcomes	Remarks
<p>a. Relational Databases Concepts (6 hours)</p>	<ul style="list-style-type: none"> • Describe the basic concepts of relational databases. • Create a simple relational database. • Describe the purposes of rollback. 	<p>It includes entity, relationship, attribute, domain, index, key (such as primary key, foreign key and candidate key), and integrity (such as entity integrity, referential integrity and domain integrity). Students should be able to identify these basic elements in examples taken from everyday applications.</p> <p>Students should know how to organise data differently but sensibly in a relational database and be able to establish the required relationships among the tables.</p>

Topic	Learning Outcomes	Remarks
b. SQL (18 hours)	<ul style="list-style-type: none"> • Use SQL to maintain a simple relational database, and manipulate its data or retrieve the required information in at most three tables. 	<p>Skills involved include:</p> <ul style="list-style-type: none"> • modify the structure of the tables • add, delete and modify the data in the tables • view, sort, and select contents by filtering, and create different views • use appropriate operators and expressions such as arithmetic operators and expressions, comparison operators, logical operators and the in, between and like operators. to perform specific operations • use simple built-in functions such as aggregate and string functions • perform queries on multiple tables, including the use of equi-join, natural join and outer join • perform sub-queries (for one sub-level only)
c. Database Design Methodology (14 hours)	<ul style="list-style-type: none"> • Be aware of the different types of relationships among entities in a relational database. • Analyse simple scenarios in business, education or other fields and create simple ER diagrams involving binary relationship only in designing databases. 	<p>The resolution of many-to-many relationship into multiple one-to-many relationships should also be introduced.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Explain the concepts of data redundancy and discuss the methods or measures used to reduce data redundancy. • Describe the needs and procedures of denormalisation. • Transform the ER diagrams to tables in relational databases. • Use access rights to achieve data privacy. 	<p>Students should be able to reduce data redundancy through normalisation up to Third Normal Form (3NF).</p>

Symbols used in ER 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	

B. Web Application Development

Introduction

This option is designed to provide students an understanding of the basic concepts network services and client-server communications, and the knowledge and skills associated with the set up and implementation of a simple network with different services. It provides them with useful and practical knowledge related to web authoring and web application development. This includes the learning of writing scripts on client-side and server-side, and developing simple web applications.

Students should have acquired the necessary knowledge and skills in the Compulsory Part – module C *Internet and its Applications* before progressing to this option.

All the sub-topics under topics (a) “Network Services and Implementation” and (b) “Web Programming and Applications” in the present option are designed as extensions of topics (a) “Networking and Internet Basics” and (c) “Elementary Web Authoring” in the Compulsory Part – module C *Internet and its Applications*.

Learning Objectives

Students will learn about:

- the basic concepts of network services;
- the implementation of a simple computer network;
- web authoring and publishing;
- writing client-side and server-side scripts; and
- developing simple web applications.

The time allocation for the option is about 38 hours.

Details

The option *Web Application Development* comprises two topics: “Network Services and Implementation” and “Web Programming and Applications”. Details of the two topics are summarised below:

Topic	Learning Outcomes	Remarks
a. Network Services and Implementation (14 hours)		
i. Basic concepts of client-server communication	<ul style="list-style-type: none"> • Know the basic concepts of client-server communication. • Know the roles of client and server as two network programs in a network. • Describe common network services. 	<p>This includes the concepts of request and response, including port number in TCP, GET and POST request in HTTP.</p> <p>The common network servers include the Dynamic Host Configuration Protocol (DHCP) server, domain controller, file server, proxy server, web server and database server, gateway, etc.</p>
ii. Basic network implementation	<ul style="list-style-type: none"> • Set up simple Ethernet and wireless networks. • Share various resources among the networked computers/stations. • Set folder/ file-sharing permissions, including read, write and execute rights, etc. 	<p>Examples of simple networks include home networks and ad hoc networks in small exhibitions and special events.</p> <p>The resources include files, printers and Internet connection, etc.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Set up simple network services. 	Examples of network services include web service and database service.
b. Web Programming and Applications (24 hours)		
i. Web authoring and publishing	<ul style="list-style-type: none"> • Edit web pages. • Publish content on web. 	<p>Students should be able to edit HTML code of web pages and apply a consistent look and style across a set of web pages through Cascading Style Sheets (CSS).</p> <p>Students should know different ways for publishing content on the web. For example, building a website and posting content through a web-based content management system.</p>
ii. Web programming and applications	<ul style="list-style-type: none"> • Understand the difference between server-side and client-side technologies. • Create client interface. 	<p>Students should be able to create client interface in form of web page and application.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Demonstrate simple client-side and server-side scripts. • Develop a simple web application. • Be aware of the new trends in web application development. 	<p>Examples:</p> <ul style="list-style-type: none"> • Process input data from clients (e.g. calculation on quadratic equation) • Simple data validation (e.g. check the range of numeric input) • Retrieve and update data from a single-table database • Use of Cookies for storing of user information <p>Students should be able to integrate client-side and server-side scripts to develop the application.</p> <p>Examples:</p> <ul style="list-style-type: none"> • Search engine on restaurants in a shopping mall • Homework checklist for a class • Personal weight management system

C. Algorithm and Programming

Introduction

This option is designed to provide students with the further programming concepts, and to develop their problem-solving skills through a systematic approach to algorithm design and programming, as well as to improve their logical thinking and critical thinking skills. They will be able to design and develop computer programs independently for solving problems and interacting with physical devices. Students will also be exposed to various principles and techniques of algorithm design. This understanding will offer students a wider choice from which to think and select an appropriate approach to solve a specific problem in future.

Students should have acquired the necessary knowledge and skills in the Compulsory Part – module D *Computational Thinking and Programming* before progressing to this option.

The sub-topics “Design of solution and Implementation” and “Testing and debugging” under topic (a) “Programming” in the present option are designed as extensions of sub-topics “Problem Formulation and Problem-Solving”, “Algorithm design”, “Program Development” and “Program Testing and Debugging” in module D *Computational Thinking and Programming*.

Learning Objectives

Students will learn about:

- the importance of good programming skills and good programming styles;
- the importance of formulating appropriate algorithms in solving problems;
- creative design and development of computer programs; and
- how to write programs to solve real-life problems.

The time allocation for the option is about 38 hours.

Details

The option *Algorithm and Programming* comprises two topics: “Programming” and “Applications of Programming in Real Life”. Details of the two topics are summarised below:

Topic	Learning Outcomes	Remarks
a. Programming (32 hours)		
i. Design of solution and implementation (28 hours)	<ul style="list-style-type: none"> • Select appropriate data types for a solution. • Review algorithms with flowcharts or pseudocode. • Apply algorithms of counting, accumulating, swapping, searching, sorting and merging in writing programs. • Choose an appropriate algorithm for a task. 	<p>The data types include simple data types, structured data types and user-defined data types. Simple data types have been discussed in the Compulsory Part.</p> <p>The search algorithms include linear search and binary search. The sorting algorithms include bubble sort, insertion sort and selection sort. Students should realise that there are other faster sorting algorithms, say merge sort and quick sort. Merging involves only two arrays of data at one time.</p> <p>Students should be aware that a problem may be solved by several different algorithms. The selection of algorithms sometimes depends on their complexities and data structures as well as their trade-off. Students should evaluate the algorithm with respect to efficiency, correctness and appropriateness for a task.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> • Apply various basic constructs in a solution. • Apply control structures in a solution. • Construct lists, stacks and queues in terms of arrays. • Manipulate text files through file-handling statements. • Implement parameters passing in manipulating sub-programs. • Realise the importance of good programming styles. 	<p>Students should be able to use global variables, local variables, constants, assignment statements, input statements, output statements, arithmetic operators, string operators, Boolean operators and Boolean logic, and operations of the operators including precedence and association.</p> <p>Sequence, selection and iteration have been introduced in the Compulsory Part. Nested loop is required.</p> <p>Students should be able to create and manipulate linear linked lists, stacks and queues in terms of arrays.</p> <p>The manipulation involves file updating statements to delete, insert, append and amend records.</p> <p>The sub-programs are called by two parameters passing methods: call by value and call by reference.</p> <p>Students should develop the habit of using meaningful variable names, comments, annotations, space and indentation.</p>

Topic	Learning Outcomes	Remarks
ii. Testing and debugging (4 hours)	<ul style="list-style-type: none"> • Appraise the use of structured programming to design, implement, and debug errors. • Interpret errors. • Apply both manual methods and software debugging tools to program debugging. 	<p>The numerical errors included are rounding errors, truncation errors, overflow errors and underflow errors. Other errors are syntax errors, logical errors and run-time errors.</p> <p>Use stubs, flags, break points and program traces with test data set for debugging.</p>
b. Applications of Programming in Real Life (6 hours)		
	<ul style="list-style-type: none"> • Use extended programming modules or libraries in writing programs to interact with physical devices. 	<p>Students should be able to use extended modules or libraries for capturing data from sensors (e.g. light sensor and accelerometer) and controlling specific devices (e.g. motor).</p> <p>Details of extended modules or libraries are not required.</p>

Topic	Learning Outcomes	Remarks
	<ul style="list-style-type: none"> <li data-bbox="817 263 1420 336">• Use event handlers in writing event-driven programs. <li data-bbox="817 523 1420 667">• Construct simple programs on physical devices by using features/components of physical devices like speech recognition and accelerometer. 	<p data-bbox="1438 263 2033 406">Specific events include user actions (e.g. pressing a button) and sensor values (e.g. the reading from the light sensor is over a defined value).</p> <p data-bbox="1438 448 1989 483">Details of event handlers are not required.</p> <p data-bbox="1438 523 2002 667">Examples include generating a text display by speech recognition, controlling the movement of motors and detecting motion by accelerometer.</p>