COMPUTER APPLICATIONS
SYLLABUS
Secondary One to Four

Implementation starting with
2019 Secondary One Cohort

Ministry of Education
SINGAPORE

CONTENTS

SECTION 1: INTRODUCTION

Value of Computer Applications
Development of the Syllabus
Curriculum Framework
Aims of the Syllabus
21st Century Competencies (21CC) in Computer Applications

1. INTRODUCTION

Value of Computer Applications

The invention and advancement of computers has revolutionized the way we live, work and learn. Having the knowledge and skills to tap on the power of computers makes a person more efficient. It is, however, important that the use of computers be grounded on sound ethics.

Computer Applications (CPA) was introduced in 1994 as a compulsory subject for secondary students in the Normal (Technical) course. Since its introduction, the syllabus has undergone several revisions to ensure its relevance and appeal to students.

The value of CPA is fourfold:

a) Promotes digital literacy. CPA prepares students to be technologically adept and to contribute effectively in a society that is increasingly driven by ICT. Students learn to be responsible, confident and creative users of technology who can participate and thrive in a digital world.
b) Engages students and prepares them for future studies. The hands-on and interactive nature of the subject effectively engages students in their learning. Through engaged learning, students are equipped with a wide range of necessary ICT skills that prepare them to meet the demands of other subjects in secondary school and at post-secondary institutes.
c) Supports the development of 21st Century Competencies (21CC). Through CPA, students acquire skills such as critical thinking, information processing and effective communication.
d) Develops computational thinking skills. Since 2012, the CPA syllabus has included programming where students learn basic computational thinking skills such as algorithmic thinking and decomposition through creating animations and games.

Development of the Syllabus

To support the development of the 2019 CPA syllabus, a syllabus review was conducted between 2016 and 2017. The review involved focus group discussions with teachers and students, a literature review of local and international computer syllabuses and the forming of a syllabus review committee.

The design considerations of the 2019 CPA syllabus are as follows:

- alignment with the Computing Curriculum Framework;
- incorporation of basic computational thinking skills;
- adoption of the 'learning by doing' and 'problem-driven' approaches to help students develop problem-solving skills and relate their learning to daily life; and
- provision of opportunities to develop 21CC.

Curriculum Framework

The design of the 2019 CPA syllabus is guided by the Computing Curriculum Framework which was revised in 2017. See Figure 1. It consists of the following:

- Vision statement for computing education
- Dimensions of computing
- Core Concepts of computing
- Components of computational thinking (CT)
- Practices of computing practitioners and professionals

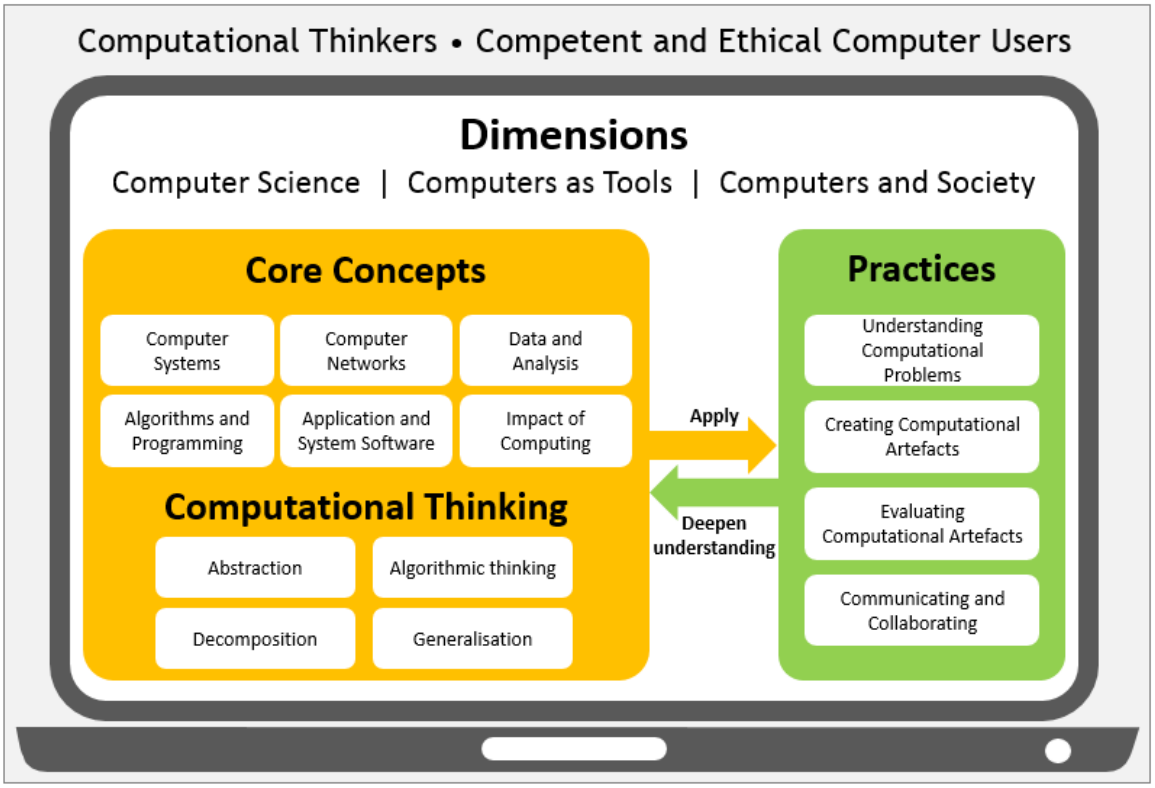


Figure 1: Computing Curriculum Framework (2017)

An important aspect of the framework is the relationship between the Core Concepts, Computation Thinking and Practices: Core Concepts and Computational Thinking are applied through the Practices, and the Practices will in turn deepen one's understanding of the Core Concepts.

Table 1 shows the alignment between the CPA topics and the Core Concepts in the Computing Curriculum Framework.

Table 1: Alignment of CPA topics with the Core Concepts in the Framework

Core Concepts	**Topics in CPA**
Computer Systems	Computer hardware (e.g. input, output, and storage devices), system specifications and troubleshooting
Computer Networks	Types of networks, network devices and internet applications
Data and Analysis	Data processing, validation and analysis using spreadsheet software
Algorithms and Programming	Visual programming, algorithms and flowcharts
Application and System Software	Operating systems and application software (word processing, graphics, presentation, spreadsheet and programming software)
Impact of Computing	Responsible use of computers, copyright issues, computer crimes and impact of ICT

Table 2 shows the alignment between the CPA tasks and the Practices in the Computing Curriculum Framework.

Table 2: Alignment of tasks in CPA with the Practices in the Framework

Practices	**Tasks in CPA**
Understanding Computational Problems	Students understand and identify key information about a complex problem.
Creating Computational Artefacts	Students design and create computational artefacts such as graphics, presentations, animations and games.
Evaluating Computational Artefacts	Students test, evaluate and improve computational artefacts (incl. debugging and refining programs).
Communicating and Collaborating	Students work collaboratively in pairs or small groups to solve problems and describe / document their solutions.

Aims of the Syllabus

Through the CPA curriculum, students learn to use various software applications as well as programming concepts. Students also gain awareness of the ethical, legal and security issues relating to the use of computers. Specifically, the aims of the syllabus are to:

1) acquire skills in using a variety of computer application software and hardware to accomplish tasks and communicate ideas;
2) appreciate the ethical, legal and security issues relating to the use of computers and ICT in society;
3) recognise the impact of ICT on society and people; and
4) develop basic computational thinking and problem-solving skills.

21st Century Competencies (21CC) in Computer Applications

The MOE 21CC framework (Figure 2) spells out the important competencies that students need to thrive in the 21st century. These competencies are anchored on a set of enduring values.

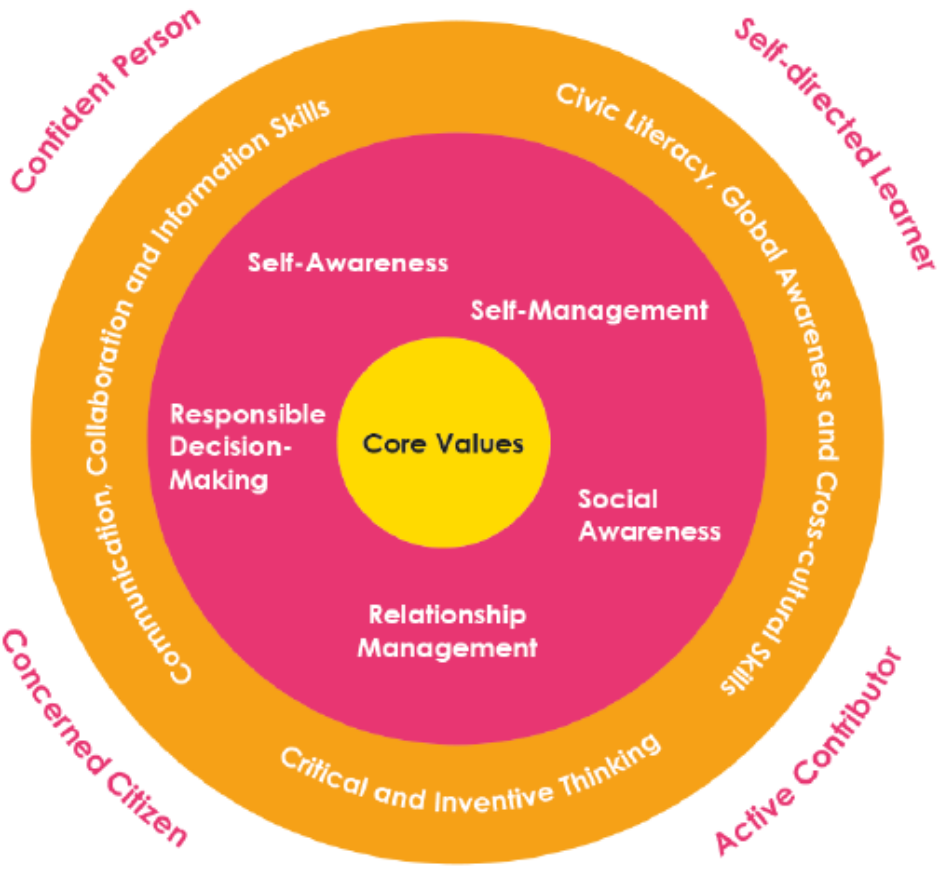


Figure 2: Framework for 21CC and Student Outcomes

At the heart of the framework are the **Core Values** that underpin the learning in the total curriculum. These values (respect, responsibility, resilience, integrity, care and harmony) define a person's character and shape beliefs, attitudes and actions of a person.

The middle ring signifies the **Social and Emotional Competencies** – skills necessary for students to recognise (self-awareness) and manage (self-management) their emotions, develop care and concern for others (social awareness), make responsible decisions (responsible decision-making), establish positive relationships, as well as to handle challenging situations effectively (relationship management).

The outer ring of the framework represents the emerging **21st Century Competencies** necessary for the globalised world we live in. These are: Civic Literacy, Global Awareness and Cross-Cultural Skills (CGC); Critical and Inventive Thinking (CIT); and Communication, Collaboration and Information skills (CCI).

The CPA curriculum provides opportunities for the development of 21CC. Core Values such as responsibility and resilience are fostered through the 'problem-driven' pedagogical approach (see Section 3). Social and Emotional Competencies are developed when students work collaboratively on different tasks. Table 3 maps CPA's competencies and attitudes to relevant emerging 21st Century Competencies.

Table 3: Development of 21CC in N(T) level CPA

Critical and Inventive Thinking (CIT)		**CPA Competencies and Attitudes**
Standards	**21st Century Competencies Benchmarks (By end of S4)**	**Understanding Computational Problems** **Creating and Evaluating Computational Artefacts**
CIT 1: Explores possibilities and generates ideas	1.1d: The student is able to generate ideas and explore different pathways that lead to solutions.	Ability to brainstorm ideas to solve problems and explore different plausible solutions.
CIT 2: Exercises sound reasoning, decision-making and metacognition	2.2d: The student is able to suspend judgement, reassess conclusions and consider alternatives to refine his/ her thoughts, attitudes, behaviour and actions.	Ability to debug, evaluate and refine computational artefacts.
CIT 3: Manages complexities and ambiguities	3.1d: The student is able to identify essential elements of complex tasks, stay focused on them, take on diverse roles and persevere when they encounter difficulties and unexpected challenges. 3.2d: The student is able to manage uncertainty and adapt to diverse demands and challenges in new and unfamiliar contexts.	Ability to: • identify the key information about a complex task; • analyse and break down a complex computational problem into manageable parts; and • persevere in creating computational artefacts despite challenges (e.g. not giving up when their computer programs do not work).

Communication, Collaboration and Information (CCI)		CPA Competencies and Attitudes
Standards	**21st Century Competencies Benchmarks (By end of S4)**	**Communicating and Collaborating**
CCI 1: Communicates and collaborates effectively	1.1d: The student is able to convey complex information and ideas coherently and clearly to create impact for specific purposes and contexts. 1.2c: The student is able to interact with others to construct knowledge, and new understandings and ideas.	Ability to: • explain and justify the appropriateness of their computational designs and choices; • describe the features and operation of their computational artefacts; and • consider different perspectives when working with others to solve computational problems.
CCI 2: Manages, creates and shares digital information thoughtfully, ethically and responsibly	2.1c: The student is able to refine search results, organise information systematically and manage information sensitively, while abiding by copyright regulations and minimising security risks in the handling of information. 2.3c: The student is able to modify and integrate varied media elements to construct a digital artefact, while making informed choices with regard to how to use information ethically. 2.5c: The student is able to project a positive online presence and manage his/ her online reputation.	Ability to: • perform online searches efficiently using an internet search engine (including advanced search options such as searching for images by usage rights); • create interactive slide presentations with text, graphics, animations, sound and/or video; • show awareness of copyright issues; and • understand the dangers of the internet (including cyber-wellness) and apply safety and security measures when using the Internet.
Civic Literacy, Global Awareness and Cross-cultural Skills (CGC)		**CPA Competencies and Attitudes**
Standards	**21st Century Competencies Benchmarks (By end of S4)**	**Understanding Computational Problems**
CGC 1: Aware of community and national issues and plays a part to improve the community and nation	1.1d: The student is able to discuss issues that affect the culture, socio-economic development, governance, future and identity of Singapore, and use evidence to support their viewpoints.	Ability to recognise and state how the use of computers and ICT has impacted society and the way people live and work.

SECTION 2: CONTENT

2. CONTENT

Overview of Content

This syllabus consists of six modules: Computer Fundamentals, Media Elements, Document Processing, Spreadsheets, Interactive Multimedia Communication and Animation and Game Making.

The overview of the content is shown below.

Module	Topics
1. Computer Fundamentals (CPF)	1.1 Computer systems 1.2 Responsible use of computers 1.3 Computer networks and communication 1.4 Impact of ICT
2. Media Elements (MEL)	2.1 Vector graphics 2.2 Raster graphics 2.3 Audio and video
3. Document Processing (DOP)	3.1 Body text 3.2 Page properties 3.3 Graphics and text boxes 3.4 Working with tables
4. Spreadsheets (SST)	4.1 Data display 4.2 Data processing 4.3 Data validation and analysis
5. Interactive Multimedia Communication (IMC)	5.1 Components and applications of interactive multimedia 5.2 Storyboarding 5.3 Creation of interactive slide presentations
6. Animation and Game Making (AGM)	6.1 Visual programming language 6.2 Planning 6.3 Programming and debugging 6.4 Documentation

Module 1: Computer Fundamentals (CPF)

A computer system is made up of both hardware and software that allow it to perform a wide range of useful tasks. The development of computer networks, particularly the internet, has extended the usefulness of computer systems by allowing users to share resources as well as to communicate and collaborate remotely. However, this advancement has also exposed users to malicious activities.

In this module, students are introduced to the essential ideas of computer fundamentals through the following units of study:

- Computer systems
- Responsible use of computers
- Computer networks and communication
- Impact of ICT

The Essential Questions for this topic are:

- What makes up a computer system?
- How do we use computers responsibly and safely?
- What are computer networks and how do they benefit computer users?
- How has the use of ICT impacted our lives?

Topic	Learning Outcomes	Sec
[1.1] Computer systems		
❑ Hardware and system specifications	[1.1.1] Define computer hardware.	1
	[1.1.2] Name the key components of a computer system such as its processor, memory and secondary storage.	1
	[1.1.3] State the difference between volatile and non-volatile memory and give examples of each.	2
	[1.1.4] Compare computers in terms of system specifications such as processor speed, memory capacity and secondary storage capacity.	3
❑ Input-Process-Output	[1.1.5] Distinguish between input, process and output operations.	1
	[1.1.6] Identify the input, process and output of a computer application.	1
	[1.1.7] Understand that meaningful information is output only after a computer has processed the correct input data.	1
❑ Input and output devices	[1.1.8] Give examples of common input and output devices.	1
❑ Software	[1.1.9] Define computer software.	1
	[1.1.10] Describe the functions of operating systems.	1
	[1.1.11] Give examples of common application software and describe their functions.	1
	[1.1.12] Distinguish between operating systems and application software.	1
	[1.1.13] Give examples of common features of graphical user interfaces such as windows, icons, menus and pointers.	1

Topic	Learning Outcomes	Sec
	[1.1.14] State the benefits of file compression such as reducing file size and making transfer easier by combining multiple files into a single file.	2
❑ Storage	[1.1.15] Compare the sizes of data units such as bits, bytes, kilobytes, megabytes, gigabytes, terabytes and petabytes.	2
	[1.1.16] Distinguish between primary and secondary storage.	2
	[1.1.17] Give examples of secondary storage media.	2
[1.2] Responsible use of computers		
❑ Troubleshooting	[1.2.1] Give examples of simple troubleshooting techniques such as rebooting and ending processes.	1
❑ Care and prevention	[1.2.2] Give examples of ways to properly handle and care for computer equipment.	1
	[1.2.3] Give examples of measures to prevent data loss such as making backups for possible recovery in case the originals are damaged.	1
	[1.2.4] Give examples of ways in which computer equipment can be damaged.	1
❑ Copyright and internet safety	[1.2.5] Show awareness of copyright issues.	1
	[1.2.6] Understand the dangers of the internet (including cyber-wellness).	1
	[1.2.7] Give examples of safety and security measures to follow when using the internet.	1
❑ Malicious Software	[1.2.8] Give examples of types of malware.	3
	[1.2.9] Describe the effects of malware.	3
	[1.2.10] Give examples of measures to protect computers against malware.	3
❑ Computer crimes	[1.2.11] Give examples of computer crimes.	3
	[1.2.12] Give examples of measures to prevent computer crimes.	3
❑ Personal data	[1.2.13] Show awareness of data privacy.	3
[1.3] Computer networks and communication		
❑ Purpose	[1.3.1] State and understand that computers in a network can facilitate communication and sharing of resources such as documents, hardware and software.	3
❑ Network hardware	[1.3.2] Give examples of common computer network devices (such as network interface cards, wireless access points, routers and modems) and state their purposes.	3
❑ Types of networks	[1.3.3] Differentiate between local area networks (LANs) and wide area networks (WANs) based on their geographical scope.	3
	[1.3.4] Understand the difference between intranets and the internet.	3
	[1.3.5] Understand the difference between wired and wireless communications.	3
❑ Internet applications	[1.3.6] Send and reply emails with one or more attachments.	2
	[1.3.7] Perform online searches efficiently using an internet search engine (including advanced search options such as searching for images by labelled usage rights).	2

Topic	Learning Outcomes	Sec
	[1.3.8] State examples of how the internet can be used for communication.	4
	[1.3.9] State the advantages of using web pages as a form of communication.	4
	[1.3.10] Collaborate on shared resources through the internet.	4
	[1.3.11] Create custom web-based forms for data collection.	4
[1.4] Impact of ICT		
❑ Role and use	[1.4.1] Describe situations where computers are used in society.	4
	[1.4.2] State how the use of computers has affected the way people live and work.	4

Module 2: Media Elements (MEL)

Media elements are widely used in many applications to engage the target audience and communicate messages effectively. Common media elements include photographs, graphics, sounds and videos. Although there are dedicated software for specific types of media elements, many application software also come with built-in features that allow users to edit multiple types of media elements.

In this module, students will learn how to create and edit different types of media elements through the following units of study:

- Vector graphics
- Raster graphics
- Audio and video

The Essential Questions in this module are:

- What are the common types of media elements?
- How do we create and edit different types of media elements?

Topic	Learning Outcomes	Sec
[2.1] Vector graphics		
❑ Nature	[2.1.1] Explain that vector graphics are created using nodes and paths.	1
	[2.1.2] State that vector graphics can be resized without loss of details.	1
❑ Basic tools and skills	[2.1.3] Create drawings using objects such as lines, curves, text, ellipses, rectangles and polygons/stars.	1
	[2.1.4] Move, resize, rotate, skew and flip objects.	1
	[2.1.5] Duplicate/copy and delete objects.	1
	[2.1.6] State that objects are arranged in a front to back order and be able to rearrange the order of objects.	1
	[2.1.7] Group multiple objects into a single object and ungroup them again.	1
	[2.1.8] Recognise common fill styles such as solid fill, gradient fill and pattern fill.	1
	[2.1.9] Set the fill of objects using a specified colour and style.	1
	[2.1.10] Set the transparency of objects such that objects underneath them are visible.	1
	[2.1.11] Set the stroke of objects using a specified colour and thickness.	1
	[2.1.12] Put text to follow the outline of an object.	1
	[2.1.13] Export vector graphics as raster graphics.	1
❑ Plan	[2.1.14] Plan how complex shapes can be formed using simpler shapes.	1
❑ Advanced tools and skills	[2.1.15] Create complex objects by using union and/or difference on simpler objects.	1
	[2.1.16] Transform one path into another path over a specified number of steps (i.e., interpolate).	1
	[2.1.17] Modify objects by manipulating their nodes and node handles directly.	1

Topic	Learning Outcomes	Sec
[2.2] Raster graphics		
❑ Nature	[2.2.1] State and recognise that raster graphics are composed of individually coloured pixels.	1
	[2.2.2] Give examples of different file formats for raster graphics and state if transparency is supported for each file format.	2
	[2.2.3] State that resizing raster graphics can result in a loss of quality.	1
	[2.2.4] Explain that the output resolution of raster graphics is measured in dots per inch (dpi) or pixels per inch (ppi) when printed on paper or displayed on a screen respectively.	2
❑ Skills	[2.2.5] Use built-in raster graphics editing functions in word processing/multimedia presentation software to enhance raster graphics.	2
[2.3] Audio and video		
❑ Audio	[2.3.1] Record and store digital voice.	2
	[2.3.2] Edit sound clips by performing trim, insert and volume control operations.	2
❑ Video	[2.3.3] Understand that frames are individual images in videos.	2
	[2.3.4] Create a video file from still images and videos with text, transitions and sound.	2
	[2.3.5] State that videos with higher frame rates can take up more space but can also appear smoother than videos with lower frame rates.	2

Module 3: Document Processing (DOP)

Documents such as letters, brochures, information sheets and reports are widely used to communicate messages and ideas. To achieve this, documents need to present the information clearly and be visually appealing to catch the attention of readers. This can be done through the appropriate use of page layout, tables, charts and graphics. At times, documents are addressed to specific persons or groups and thus need to be customised.

In this module, students will learn how to use the features of word processing software to create effective documents for different purposes through the following units of study:

- Body text
- Page properties
- Graphics and textboxes
- Working with tables

The Essential Questions in this module are:

- How do we use the features of word processing software to create visually effective documents?
- How do we use mail merge features to generate customised letters?

Topic	Learning Outcomes	Sec
[3.1] Body text		
❑ Text editing and formatting	[3.1.1] Recognise serif and sans-serif typefaces.	1
	[3.1.2] Use decorative text such as word art and drop caps to improve the appearance of a document.	1
	[3.1.3] Use the find and replace feature.	1
❑ Paragraph formatting	[3.1.4] Differentiate between paragraphs that are left-aligned, centred, right-aligned and justified.	1
	[3.1.5] Set paragraphs to be left-aligned, centred, right-aligned or justified.	1
	[3.1.6] Set line spacing to a specified value.	1
	[3.1.7] Set the spacing before or after a paragraph to specified values.	1
	[3.1.8] Insert soft breaks to start new lines without starting new paragraphs.	3
	[3.1.9] Differentiate between paragraphs which have their left, right, first-line and/or hanging indents set.	3
	[3.1.10] Set the left, right, first-line and/or hanging indents of a paragraph to specified values.	3
	[3.1.11] Use pre-set tab stops to align different lines of text (left-align only).	3
[3.2] Page properties		
❑ Page layout	[3.2.1] Recognise portrait and landscape orientations and give examples of what each orientation is used for.	1
	[3.2.2] Set the orientation of a page.	1
	[3.2.3] Set page margins to specified values.	1
	[3.2.4] Use text boxes and tables to lay out a document.	3
	[3.2.5] Give examples of scenarios where single-column or multiple-column layouts may be used.	3

Topic	Learning Outcomes	Sec
	[3.2.6] Create single-column and/or multiple-column layouts using text boxes, tables and/or the built-in columns feature.	3
❑ Page formatting	[3.2.7] Insert page borders.	2
	[3.2.8] Describe the purpose of headers and footers.	2
	[3.2.9] Insert headers and footers.	2
	[3.2.10] Insert footnotes.	2
	[3.2.11] Insert page numbers and set the starting page number to a specified value.	2
	[3.2.12] Insert or remove page breaks.	4
	[3.2.13] Use section breaks to format a document, including allowing page layout in both portrait and landscape orientation.	4
[3.3] Graphics and text boxes		
❑ DOP graphics	[3.3.1] Insert shapes such as lines, curves, ovals, rectangles, polygons and stars.	1
	[3.3.2] Move, resize, flip and rotate shapes.	1
	[3.3.3] Insert text into shapes.	1
❑ Imported graphics	[3.3.4] Import vector or raster graphics into a document.	1
	[3.3.5] Move, resize, flip and rotate imported graphics.	1
	[3.3.6] Crop imported raster graphics and discard the cropped area.	1
❑ Charts	[3.3.7] Embed charts created using spreadsheet software.	3
❑ Watermark	[3.3.8] Create picture or text watermarks.	3
❑ Text boxes	[3.3.9] Create text boxes and import text into them.	3
	[3.3.10] Link text boxes so that text flows from one to another in a specified order.	3
	[3.3.11] Modify the borders of a text box.	3
❑ Text wrapping	[3.3.12] Set the wrapping style of a text box or graphic.	3
[3.4] Working with tables		
❑ Appearance and alignment	[3.4.1] Insert or remove table rows and columns.	1
	[3.4.2] Format and shade table cells.	1
	[3.4.3] Merge table cells.	1
	[3.4.4] Modify table cell borders.	1
	[3.4.5] Set tables to be left-aligned, centred or right-aligned.	1
	[3.4.6] Align text and graphics in table cells vertically and horizontally.	1
❑ Mail merge	[3.4.7] State the advantage of using mail merge.	4
	[3.4.8] Use the mail merge feature with data from a table in a word document.	4
	[3.4.9] Use the mail merge feature with data from a table in a spreadsheet.	4
	[3.4.10] Filter the recipient list when using the mail merge feature by setting criteria[1] on specified fields.	4

[1] Limited to “greater than”, “less than”, “equal to” and “not equal to”.

Module 4: Spreadsheets (SST)

Spreadsheet software are widely used to tabulate and collate data. The data can then be organised, processed and analysed easily using features such as formulas, functions and charting tools.

In this module, students will learn how to use the features of spreadsheet software to store, process, analyse and present data through the following units of study:

- Data display
- Data processing
- Data validation and analysis

The Essential Questions in this module are:

- How do we use the features of spreadsheet software to process and analyse data?
- How do we present data using the different types of charts?
- How do we validate data in spreadsheets?

Topic	Learning Outcomes	Sec
[4.1] Data display		
❑ Cell formats	[4.1.1] Set cells to use either a number, currency or percentage format with a specified number of decimal places.	1
	[4.1.2] Set cells to use a specified date format.	1
	[4.1.3] Wrap and align text in cells vertically and horizontally.	1
	[4.1.4] Use conditional formatting to change the fill and/or font colour of cells based on their contents[2].	3
❑ Charts	[4.1.5] State the purpose of different chart types such as bar charts, column charts, pie charts and line charts.	2
	[4.1.6] Create bar charts, column charts, pie charts or line charts with data from either a continuous or non-continuous range of cells.	2
	[4.1.7] Recognise that modifying a chart's data table will cause a corresponding change to the chart.	2
	[4.1.8] State the purpose of combination charts.	3
	[4.1.9] Create combination charts[3].	3
❑ Print area	[4.1.10] Set the print area of a spreadsheet to a specified range of cells.	3
	[4.1.11] Scale the width and/or height of a spreadsheet to fit a specified number of pages when printed.	3
[4.2] Data processing		
❑ Formulas	[4.2.1] Use mathematical operators (+, -, * and /) in formulas.	1
	[4.2.2] State the advantages of using cell referencing in formulas for calculations.	1
	[4.2.3] Recognise that the value of cells which use formulas will be automatically recalculated when their referenced cells are changed.	1
	[4.2.4] Change the view of a spreadsheet to display formulas.	1

[2] Limited to "greater than", "less than" and "equal to".

[3] Limited to the combination of a line chart and a column chart.

Topic	Learning Outcomes	Sec
	[4.2.5] Differentiate between absolute and relative cell referencing.	3
	[4.2.6] Use absolute and relative cell referencing.	3
❑ Functions	[4.2.7] Use functions to find the minimum, maximum, sum and average of values in a range of cells.	1
	[4.2.8] Use functions to find the mode[4] and median of values in a range of cells.	3
	[4.2.9] Use a function to get a random integer between two specified integers (both inclusive).	3
	[4.2.10] Use functions to round positive values either by rounding to the nearest whole number, by always rounding up or by always rounding down.	2
	[4.2.11] Use functions to get either the current date or the current date and time.	2
	[4.2.12] Use functions to extract a range of characters from the left or middle of some specified text.	3
	[4.2.13] Use a function to find the number of characters in some specified text.	3
❑ Conditions	[4.2.14] Use functions to count the number of cells in a range of cells that meet a specified criterion[5].	3
	[4.2.15] Use conditional if functions in formulas.	3
	[4.2.16] Use relational operators (>, >=, <, <= and =) to compare values in formulas.	3
	[4.2.17] Use a function to look up data in a specified range of cells and return the value of another cell from the first row that matches the data exactly.	4
❑ Sort and filter	[4.2.18] Sort cells in ascending or descending order based on the contents of a particular column.	3
	[4.2.19] Filter data by setting criteria[6] on a column.	4
[4.3] Data validation and analysis		
❑ Input validation	[4.3.1] Restrict input to a range of cells.	4
	[4.3.2] State why input data may need to be validated.	4
	[4.3.3] Set validation criteria for cells.	4
	[4.3.4] Display custom error messages when invalid input data is keyed in.	4
❑ What-if	[4.3.5] Determine the value needed in a cell for another cell to reach a specified target value.	4

The list of 20 examinable functions are:

MIN, MAX, SUM, AVERAGE, MODE.SNGL, MEDIAN, RANDBETWEEN, ROUND, ROUNDUP, ROUNDDOWN, TODAY, NOW, LEFT, MID, LEN, COUNT, COUNTA, COUNTIF, IF and VLOOKUP

Students should not be penalised for using functions not listed.

[4] Limited to data set with a single mode.

[5] Criterion includes whether the cell has numeric values only, is non-empty or has a specified value only.

[6] Limited to "greater than", "greater than or equal to", "less than", "less than or equal to", "equal to" and "not equal to".

Module 5: Interactive Multimedia Communication (IMC)

Presentation software is commonly used to create multimedia presentation slides. With the appropriate use of navigation features, the presentation slides can be made interactive and allow users to explore and navigate the content easily. To achieve this, the sequence of slides and the user interface should be thoughtfully planned.

In this module, students will learn how to use the features of a presentation software through the following units of study:

- Components and applications of interactive multimedia
- Storyboarding
- Creation of interactive slide presentations

The Essential Questions in this module are:

- How do we develop a storyboard for an interactive slide presentation?
- How do we create visually effective and user-friendly interactive slide presentations using presentation software?

Topic	Learning Outcomes	Sec
[5.1] Components and applications of interactive multimedia		
❑ Media elements and interactivity	[5.1.1] State that text, graphics, animation, sound and video are media elements.	1
	[5.1.2] State that multimedia content consists of multiple media elements.	1
	[5.1.3] Identify the different types of media elements used in some specified multimedia content.	1
	[5.1.4] State the advantages of using multimedia in communication.	1
	[5.1.5] Give examples of interactive use of multimedia.	1
❑ Applications	[5.1.6] State that slide presentations are an example of interactive multimedia communication.	1
	[5.1.7] Give examples of applications where interactive multimedia is used in daily life, such as a shopping mall directory.	1
[5.2] Storyboarding		
❑ Storyboarding	[5.2.1] Create storyboards to show the flow of content and layout of media elements for a proposed interactive slide presentation.	1
[5.3] Creation of interactive slide presentations		
❑ Layout and style	[5.3.1] Adjust the relative dimensions of media elements and use white space appropriately.	1
	[5.3.2] Use appropriate fonts (including serif and sans-serif typefaces) and colours to enhance readability.	1
	[5.3.3] Use appropriate font styles and sizes to distinguish between headings and body text.	1
	[5.3.4] Use appropriate backgrounds for a slide presentation to enhance contrast with media elements.	1
	[5.3.5] Use the slide master feature to achieve a consistent style and layout.	1

Topic	Learning Outcomes	Sec
❑ Use of media and interactive elements	[5.3.6] Insert media elements into a slide presentation.	1
	[5.3.7] Insert navigation buttons for moving to the first, next, previous or last slide in a slide presentation.	1
	[5.3.8] Create text or graphic hyperlinks between slides in a slide presentation.	2
	[5.3.9] Create text or graphic hyperlinks to external resources.	2
	[5.3.10] Produce interactive slide presentations based on storyboards.	1

Module 6: Animation and Game Making (AGM)

The development of animations and games is a highly creative activity. In a visual programming language, students use blocks of code to give instructions to the computer. Through the planning and creation of animations and games, students will learn about algorithms and programming. Students will also learn how to break down a computational problem into simpler parts and come up with solutions for each part.

In this module, students will learn how to use a visual programming language to develop animations and games through the following units of study:

- Visual programming language
- Planning
- Programming and debugging
- Documentation

The Essential Questions for this module are:

- How do we develop a storyboard for an animation or game?
- How do we create animations and games using a visual programming language?
- How do we represent programming instructions using flowcharts?

Topic	Learning Outcomes	Sec
[6.1] Visual programming language		
❑ Definition	[6.1.1] Recognise that visual programming languages use graphical symbols to develop programs.	2
❑ Application	[6.1.2] State that visual programming can be used to create animations and games.	2
[6.2] Planning		
❑ Purpose	[6.2.1] Describe the key ideas of a proposed animation or game.	2
❑ Storyboarding	[6.2.2] Represent the sequence of events in a proposed animation or game using words and/or sketches.	2
	[6.2.3] Identify the backdrops needed for the stage based on a storyboard.	2
	[6.2.4] Identify the sprites needed for a storyboard.	2
	[6.2.5] Recognise the need to have different costumes for a sprite.	2
[6.3] Programming and debugging		
❑ Stage	[6.3.1] State that the area where the animation or game takes place is called the stage.	2
	[6.3.2] Recognise that points on the stage can be represented using their *x* and *y* coordinates.	2
❑ Scripts	[6.3.3] State that scripts are instructions executed by an object[7].	2
	[6.3.4] Recognise that multiple scripts can be executed at the same time.	2
❑ Motion	[6.3.5] Position sprites at a specified location and orientation.	2
	[6.3.6] Move and rotate sprites.	2
❑ Control	[6.3.7] Start and stop the execution of scripts.	2

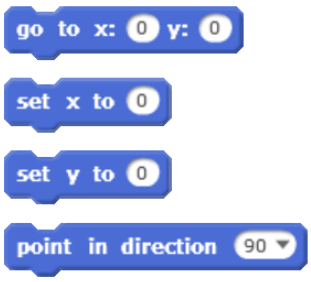
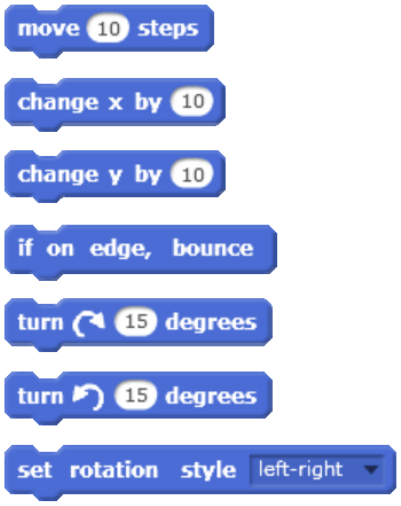
[7] Object refers to either a sprite or the stage.

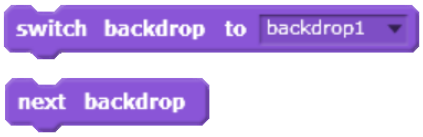
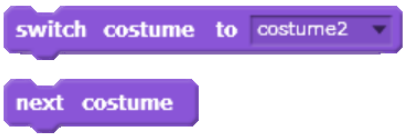
Topic	Learning Outcomes	Sec
	[6.3.8] Insert wait time between the execution of two instructions.	2
	[6.3.9] Use basic loops (repeat, forever).	2
❑ Appearance	[6.3.10] Change the brightness of the stage.	2
	[6.3.11] Insert additional backdrops to the stage by choosing from the library, importing from a file or drawing with the built-in editor.	2
	[6.3.12] Switch between the stage's backdrops.	2
	[6.3.13] Create and name sprites.	2
	[6.3.14] Change the brightness, transparency and size of sprites.	2
	[6.3.15] Insert additional costumes for a sprite by choosing from the library, importing from a file or drawing with the built-in editor.	2
	[6.3.16] Switch between a sprite's costumes.	2
❑ Input and Output	[6.3.17] Show and hide sprites.	2
	[6.3.18] Display text as either a speech or thought bubble.	2
	[6.3.19] Play sounds for an object.	2
	[6.3.20] Display and hide the values of variables.	3
	[6.3.21] Prompt for and accept text input.	3
❑ Events	[6.3.22] Send a message to trigger other objects to start executing scripts.	2
	[6.3.23] Set key presses and/or mouse clicks to trigger actions.	2
❑ Variables	[6.3.24] State that the purpose of variables is to store values.	3
	[6.3.25] Create and name variables.	3
	[6.3.26] Initialise and update the values of variables.	3
❑ Conditions	[6.3.27] Use conditional instructions (if and if-else) [8].	3
	[6.3.28] Use conditional loops (repeat-until) [8].	3
	[6.3.29] Use relational operators (>, < and =) in conditional instructions and/or loops.	3
	[6.3.30] Use contact between sprites and/or coloured areas of objects in conditional instructions and/or loops.	3
❑ Operators	[6.3.31] Generate and use random numbers in scripts.	3
	[6.3.32] Use mathematical operators (+, -, * and /) in scripts.	3
❑ Evaluation	[6.3.33] Identify and correct errors in scripts written using a visual programming language.	2
[6.4] Documentation		
❑ Flowcharts	[6.4.1] Identify common flowchart symbols.	2
	[6.4.2] Represent the instructions executed by an object using flowcharts.	2
❑ Game instructions	[6.4.3] Write game instructions on how a game is to be played.	3

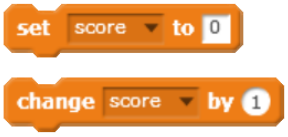
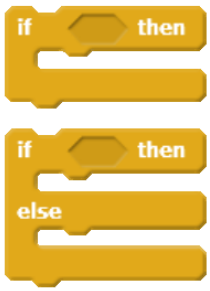
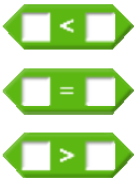
Please see Table 4 for the list of examinable Scratch blocks.
Students should not be penalised for using blocks not listed.

[8] Limited to the use of one condition per conditional instruction.

Table 4: List of examinable Scratch blocks

Topic	Learning Outcomes	Blocks
[6.3] Programming and debugging	Students should be able to	
❑ Motion	[6.3.5] Position sprites at a specified location and orientation.	
	[6.3.6] Move and rotate sprites.	
❑ Control	[6.3.7] Start and stop the execution of scripts.	 (Note: The stop block is not always required to stop execution of scripts)
	[6.3.8] Insert wait time between the executions of two instructions.	
	[6.3.9] Use basic loops (repeat, forever).	

Topic	Proposed Learning Outcomes	Blocks
❑ Appearance	[6.3.10] Change the brightness of the stage.	
	[6.3.12] Switch between the stage's backdrops.	
	[6.3.14] Change the brightness, transparency and size of sprites.	
	[6.3.16] Switch between a sprite's costumes.	
❑ Input and Output	[6.3.17] Show and hide sprites.	
	[6.3.18] Display text as either a speech or thought bubble.	
	[6.3.19] Play sounds for an object.	

Topic	Proposed Learning Outcomes	Blocks
	[6.3.20] Display and hide the values of variables.	
	[6.3.21] Prompt for and accept text input.	
❑ Events	[6.3.22] Send a message to trigger other objects to start executing scripts.	
	[6.3.23] Set key presses and/or mouse clicks to trigger actions.	
❑ Variables	[6.3.26] Initialise and update the values of variables.	
❑ Conditions	[6.3.27] Use conditional instructions (if and if-else) [9].	
	[6.3.28] Use conditional loops (repeat-until) [9].	
	[6.3.29] Use relational operators (>, < and =) in conditional instructions and/or loops.	

[9] Limited to the use of one condition per conditional instruction.

Topic	Proposed Learning Outcomes	Blocks
	[6.3.30] Use contact between sprites and/or coloured areas of objects in conditional instructions and/or loops.	
❑ Operators	[6.3.31] Generate and use random numbers in scripts.	
	[6.3.32] Use mathematical operators (+, -, * and /) in scripts.	

SECTION 3: PEDAGOGY

Pedagogical Considerations
Pedagogical Approaches
Performance Tasks

3. PEDAGOGY

Pedagogical Considerations

This section elaborates on the considerations made in the selection of pedagogical approaches and teaching strategies for CPA.

Provision of authentic contexts

Authentic learning is the pedagogy recommended by the N(T) Step Curriculum. The provision of authentic contexts means that learning activities should mirror real-world tasks. This promotes higher levels of engagement as students are required to actively apply concepts, skills and knowledge to create computational artefacts (e.g. setting up a spreadsheet to analyse test results) to solve real-world problems. This would also help students to build a foundation of knowledge by providing opportunities for them to apply what they have learnt in different contexts.

Development of problem-solving skills

To develop problem-solving skills, the pedagogical approaches and strategies should provide students with opportunities to solve a range of problems of varying difficulties and contexts. By working on these problems, students can apply and develop a set of problem-solving skills that are useful in future studies and life.

Matching students' learning profiles

For learning to be effective, teachers should design appropriate learning experiences for students after they have understood and considered the learners' profiles. Teachers will then be able to explore a variety of strategies to help their students achieve the intended learning outcomes.

Alignment with Computing Curriculum Framework

Besides creating and evaluating computational artefacts, the pedagogical approaches and strategies should allow students to be engaged in other practices outlined in the Computing Curriculum Framework. These include understanding computational problems and collaborating to solve computational tasks.

Pedagogical Approaches

The central pedagogical approaches adopted for CPA are the 'learning through doing' and 'problem-driven' approaches. See Table 5 for the key features.

Table 5: Key features of 'learning through doing' and 'problem-driven' approaches

Learning through Doing	Problem-driven
Students design and create computational artefacts.	Students work on problems which are based on authentic contexts.
Students work collaboratively to design and generate solutions to tasks/problems.	Students understand and identify key information from the description of a computation problem.
Students examine computer programs (i.e. lines of codes) to identify bugs and correct them.	Students solve problems systematically by using the Stop-Think-Act-Review approach.

Teachers should use and adapt Teaching Actions that support these two central pedagogical approaches after taking into consideration the learning needs of their students. Table 6 provides examples of Teaching Actions, most of which are adopted from the Singapore Teaching Practice (STP)[10].

Table 6: Teaching Actions applicable to teaching of CPA

Teaching Areas	Teaching Actions	Examples of how it can be used in the classroom
Providing Clear Explanation	**Demonstration** Teachers demonstrate a 'walk through' of a new skill during which students learn by observing.	• Teachers demonstrate a new skill (e.g. how to perform mail merge) to students.
	Model Thinking Aloud Teachers make thinking visible by verbalising and making explicit their thinking so that students can follow the teachers' thought processes.	• Teachers think out loud and verbalise their thought processes (e.g. "Which basic shapes are needed to create this drawing?", "Which backdrop should appear if the player wins the game?"). • Teachers step through a program line-by-line to explain how each line of code is executed (i.e. code tracing).
Using Questions to Deepen Learning	**Initiate-Response-Feedback Chains** Teachers use questions to elicit, probe and scaffold students' thinking.	• Teachers ask questions such as "What makes you say that?" after students have given a response to help students identify the basis for their thinking as they elaborate on the reasoning behind their responses.

[10] The STP covers different aspects of teaching which teachers can adopt for the teaching of CPA. Visit https://opal.moe.edu.sg/stp for more information.

Teaching Areas	Teaching Actions	Examples of how it can be used in the classroom
Encouraging Learner Engagement	**Explore, Engage, Apply** Teachers design learning activities that are meaningful and relevant to students.	• Use of kinaesthetic and unplugged activities to explain computer or programming concepts. • Students can act out the actions indicated in by a program script. This is especially useful for scripts with motion blocks.
	Engagement through Collaboration and Interactivity Teachers assign students to work collaboratively in pairs. One student (the driver) has control of the keyboard and mouse, while the other (the navigator) looks at the big picture and provides comments. Students are to switch roles from time to time.	• In pair drawing using a graphics software, the driver creates the drawing while the navigator provides feedback (e.g. accuracy, proportionality of the drawing objects). • In pair programming, the driver writes the program while the navigator considers the requirements and checks for errors.
Facilitating Collaborative Learning	**Think-pair-share** Students first think through a problem alone and then discuss in pairs. This is followed by consolidation led by teacher with the whole class.	• Students consider possible solutions to a theory question (e.g. how to take proper care of computers) and discuss answers with a partner followed by consolidation by teacher.
Instructional Strategies for Teaching Programming	**Predict and Compare** Students predict the outcome of a program/process, and compare with the actual outcome.	• Students predict behaviour of a sprite based on given scripts.
	Debugging programs Teachers provide buggy scripts for students to find and correct the errors.	• Students identify and correct the error(s) in a buggy program, e.g. the score in a game is not updated properly.

Performance Tasks

To reinforce and extend students' learning, teachers could use the following three types of tasks developed by CPDD. The tasks, in increasing levels of difficulty, are:

(a) Skill-building tasks are closely guided tasks that aim to develop understanding of basic concepts and skills within a single module. Most of these tasks use step-by-step instructions to guide students towards the learning objectives. Through these tasks, students develop the basic competencies need to take on the more complex problem sets.

(b) Problem sets are integrative tasks set within authentic contexts. Students will apply the skills from different modules to solve the problem sets. For example, a Secondary Two animation problem set may require students to first create the characters using graphics software learnt in Secondary One.

(c) Course projects are open-ended tasks that help to consolidate students' learning. Students are required to demonstrate their problem-solving abilities to create computational artefacts. Based on a given theme, students will analyse the task, decompose it into smaller parts, design their solutions and implement them. Students should be engaged in at least one course project per year (except Secondary Four). The following are the suggested course projects for each level.

- For Secondary One, the project could involve the creation of an interactive multimedia presentation for a given theme with graphics drawn using graphics software.
- For Secondary Two, the project could involve the creation of an animation with sprites that were created using graphics software.
- For Secondary Three, the project could involve the creation of a game with self-recorded sound and sprites drawn using graphics software.

To empower students to understand and complete tasks which are more complicated, teachers can introduce various problem-solving strategies. See Table 7.

Table 7: Problem-solving strategies for CPA

Strategy	Description of strategy	Examples of how it can be used in the classroom
Marking the text	Students highlight, underline and/or annotate part of the text that describes the problem.	• This is especially useful for more complex tasks where students underline key information and/or task requirements that will help them understand the problem.
Identifying subtasks/ Decomposition	Students break a problem into smaller parts whose outcomes lead to a combined solution.	• For a complicated drawing, students can identify and create the simpler parts of the drawing first before combining them together. • For game creation, students identify and work systematically on the smaller subtasks (e.g. movement of sprites, scoring, conditions for winning/losing, etc.) that make up the whole game.

Stop-Think-Act-Review (STAR)	Students use the four steps of Stop, Think, Act and Review to guide them in their approach and solution of a novel task.	• Stop - Students will pause and study the problem carefully instead of jumping into the task before understanding the requirements. • Think - Students will (1) try to understand the problem and (2) come up with a plan to solve the problem. • Act - Students will implement the plan. • Review - Students will check the solution against the requirements of the problem and make changes if necessary.

SECTION 4: ASSESSMENT

Assessment for Computer Applications
School-based Assessment
National Examination

4. ASSESSMENT

Assessment for Computer Applications

Assessment is integral to the learning process and helps students become self-directed learners. In this way, assessment is aligned to pedagogical approaches (as outlined in the previous section), curricular objectives and content. Both school-based assessment and national examinations play important and different roles in our education system.

Assessment is an important part of teaching and learning, and it is an ongoing process by which teachers gather information about students' learning to inform and support future teaching. Assessment can be categorised into the following types:

- Formative Assessment, which can be incorporated into skill-building tasks, problem sets and course projects, can be used to determine how students are progressing through certain learning outcomes during a series of learning activities. Formative assessment can be used to identify learning gaps and provide timely feedback to students on their learning as well as inform teachers on planning for future instruction. Teachers should also create opportunities for students to show that the feedback has enabled them to close their learning gaps.
- Summative Assessment, such as class tests, school and national examinations, are used at the end of a series of learning activities to determine the level of students' attainment of the desired learning outcomes. It is commonly used for placement and grading.

A balanced assessment system should have both formative and summative assessment.

School-based Assessment

School-based assessments provide opportunities for teachers to obtain information about students' level of competency and provide targeted feedback to their students. The adopted pedagogies and resources developed by CPDD provide such opportunities by getting the students to perform tasks that demonstrate their knowledge and skills. A number of examples are provided below.

(i) As students work on the skill-building tasks, problem sets and course projects, teachers can observe them and provide the necessary guidance and feedback to help them develop deeper conceptual understanding and competency.

(ii) Teachers can use Teaching Actions at appropriate junctures. For example, a teacher can use think-pair-share to get students to discuss and predict the behaviour of a sprite based on a script. From the students' responses, the teacher will get a sense of their level of understanding to better facilitate subsequent learning activities.

(iii) Teachers can get students to explain how they use the Stop-Think-Act-Review strategy. From the sharing, teachers would be able to find out the extent which students have understood the problem and determine whether further guidance should be provided.

School-based summative assessment should consist of both written and practical components. The written paper may comprise multiple-choice and short-structured questions of variable marks.

At the Upper Secondary level, the format of the assessment papers may be modelled after the format of the national examinations. The marks for school-based assessment may be used for reporting students' performance at the end of semesters.

National Examination

Assessment Objectives

The examination will assess candidates'

AO1 Knowledge and understanding of computing concepts, application software and the impact of computer technology on everyday life

AO2 Application of knowledge and understanding of computing concepts, computer application software, basic computational thinking and problem solving skills to
- analyse computational problems, and
- communicate computational solutions

AO3 Skills in using a range of application software to accomplish specific tasks

Students will demonstrate understanding of a range of computer applications and the responsible use of information. They will be able to use relevant application software to solve problems in everyday context, represent their problem solutions via data tables, flowcharts and programming statements as well as to demonstrate basic computational thinking through simple programming and debugging. They will be able to use typical office productivity application software for document processing, multimedia presentations, spreadsheet calculations and charts. Students will also be able to demonstrate their skills in user interface design, computer drawing, image, audio and video editing, as well as creating animations and games.

Scheme of Assessment

All candidates will offer Paper 1, Paper 2 and Paper 3 in the year of the examination.
All questions in all the three papers are compulsory.

Paper 1 (Written, 1 hour 15 minutes, 60 marks)
This paper will assess candidates' knowledge, understanding and application of concepts and skills in all the six modules:

- Computer Fundamentals (CPF)
- Media Elements (MEL)
- Document Processing (DOP)
- Spreadsheets (SST)
- Interactive Multimedia Communication (IMC)
- Animation and Game Making (AGM)

The paper contains two sections. Section A (20 marks) contains 20 multiple-choice questions with 4 choices per question. Section B (40 marks) contains a variable number of short-structured questions of variable mark values. There will be at least one question on representing programming instructions using flowcharts. This paper carries 30% of the subject grade and covers assessment objectives AO1 and AO2.

Paper 2 (Practical, 1 hour 30 minutes, 70 marks)

This paper will assess candidates' skills in carrying out related tasks covering three modules:

- Media Elements (MEL)
- Document Processing (DOP)
- Interactive Multimedia Communication (IMC)

The Practical Paper 2 examination will assess candidate's ability to carry out three tasks using appropriate application software: computer graphics software to create a drawing, word processing software to edit and format a given document and perform mail merge using a given source data, and presentation software to create a multimedia slide presentation with given media elements. The allotted time includes time for saving the required work in the candidates' computers. This paper carries 35% of the subject grade and covers assessment objective AO3.

Paper 3 (Practical, 1 hour 30 minutes, 70 marks)

This paper will assess candidates' skills in carrying out a series of tasks covering three modules:

- Media Elements (MEL)
- Spreadsheets (SST)
- Animation and Game Making (AGM)

The Practical Paper 3 examination will assess candidate's ability to carry out three tasks using appropriate application software: video editing software to create a video file, spreadsheet software to edit a spreadsheet and create charts, and programming software to create a game. The allotted time includes time for saving the required work in the candidates' computers. This paper carries 35% of the subject grade and covers assessment objective AO3.

Summary of details for each paper:

Paper	Mode	Duration	Weighting	Marks	Format	Modules Assessed
1	Written	1 h 15 min	30%	20	Section A: 20 MCQ	All the six modules
				40	Section B: Variable no. of Short-Structured Questions (SSQ)	All the six modules
2	Practical	1 h 30 min	35%	70	3 related tasks	MEL, DOP, IMC
3	Practical	1 h 30 min	35%	70	3 tasks	MEL, SST, AGM

Written Examination

The written paper (Paper 1) tests candidates' knowledge with understanding (~60%) and application (~40%) of concepts and skills learnt in all the six modules. Section A (20 marks) contains 20 multiple-choice questions with four options per question. Section B (40 marks) contains a variable number of short-structured questions of variable mark values.

Table of Specification for Paper 1:

Name of Module	Percentage of Paper	Marks		
		AO1	AO2	TOTAL
Section A (20 MCQ) – 20 marks				
Computer Fundamentals	~8%	~5	–	~5
Media Elements	~5%	~3	–	~3
Document Processing	~5%	~3	–	~3
Spreadsheets	~5%	~3	–	~3
Interactive Multimedia Communication	~5%	~3	–	~3
Animation and Game Making	~5%	~3	–	~3
Section B (Variable number of SSQ) – 40 marks				
Computer Fundamentals	~26%	~9	~7	~16
Media Elements	~5%	~2	~1	~3
Document Processing	~7%	~2	~2	~4
Spreadsheets	~7%	–	~4	~4
Interactive Multimedia Communication	~7%	~3	~1	~4
Animation and Game Making	~15%	–	~9	~9
TOTAL		~36	~24	60
		~60%	~40%	100%

Practical Examinations

The practical papers (Papers 2 and 3) test candidates' skills in using a range of application software to accomplish specific tasks. The maximum mark for each practical paper is 70 and each paper contributes 35% to the whole examination for Computer Applications.

Marking is based on submitted softcopies. The format of the work submitted must be in the format specified in the questions. For Paper 2 Task 1 on Media Elements (Graphics), the graphic files submitted must be in the raster format. For Paper 3 Task 1 on Media Elements (Video), the video files submitted must be in the mp4 format.

Table of Specification for Paper 2:

Module	Description of Task	Weighting	Marks
Media Elements (Graphics)	Candidates will use computer graphics software to create a given drawing according to specifications given in the question paper; fill the drawing with colours; and submit the exported drawing. The drawing will be used for the 2nd or 3rd task.	~30%	~21
Document Processing	Candidates will use word processing software to format and edit a given document according to specifications given in the question paper. Candidates are expected to demonstrate skills like importing text and images; formatting page layout with columns, tables and/or text boxes; adding headers, footers and footnotes; and submitting the required work.	~40%	~28
Interactive Multimedia Communication	Candidates will use presentation software to create a multimedia slide presentation with given media elements according to specifications described in the question paper and submit the required work.	~30%	~21
	TOTAL	**100%**	**70**

Table of Specification for Paper 3:

Module	Description of Task	Weighting	Marks
Task 1			
Media Elements (Video)	Candidates will use video editing software to create a video file with given still images and videos with text, transitions and sounds according to specifications described in the question paper; and submit the exported video.	~20%	~14
Task 2			
Spreadsheets	Candidates will use spreadsheet software to work on a given data table by creating and completing columns in the data table and performing specified tasks like ▪ using operators, formulas and functions for calculations; ▪ plotting and labelling charts; ▪ data validation; ▪ conditional formatting; ▪ sorting and manipulating data; and submitting the required work.	~40%	~28
Task 3			
Media Computing	Candidates will use programming software to work on a given game scenario to demonstrate skills in selecting suitable pictures for background pictures and characters, editing sound clips, creating scripts to animate characters, allowing user interaction through keyboard and/or mouse; and submitting the required work.	~40%	~28
TOTAL		**100%**	**70**

Specification Grid

	AO1	AO2	AO3	TOTAL
Paper 1	~18%	~12%		30%
Paper 2			35%	35%
Paper 3			35%	35%

Use of Calculator

An approved calculator may be used in Paper 1, Paper 2 and Paper 3.

Centre Infrastructure for Practical Examination

The Centre will ensure adequate hardware and software facilities to support the examination of its candidates for Paper 2 and Paper 3, which will be administered, in shifts where necessary, on the day of the examination for each of the papers. Each candidate should have the sole use of a personal computer during the examination. The hardware and software recommended by the Ministry of Education should be used for the examination. The candidates should be able to access software applications for computer graphics, word processing, presentation, spreadsheet, video editing, sound editing and programming. The Centre will be required to declare the name and version number of the software at least two years before the cohort using the software sits for the examinations.

THIS PAGE IS

INTENTIONALLY

LEFT BLANK