Singapore Examinations and Assessment Board

CAMBRIDGE
International Education

Singapore–Cambridge Secondary Education Certificate (2027)

G1 Computing
(Syllabus K127)

(First year of examination in 2027)

CONTENTS

AIMS

The aims of the G1 Computing syllabus are:

1 acquire knowledge and understanding of the concepts of computer systems, networks, application software and programming

2 develop and apply computational thinking skills such as abstraction and decomposition by creating computational artefacts

3 develop and apply media software skills by using application software

4 develop an appreciation of computing as a creative field together with an awareness of cybersecurity, emerging technology and the impact of computing

5 develop attitudes and 21CC needed to do well in computing including critical, adaptive and inventive thinking, collaboration, communication as well as perseverance in striving for accuracy and thoroughness.

ASSESSMENT OBJECTIVES

The examination will assess candidates':

AO1 knowledge and understanding of computing concepts, application software and the impact of computing

AO2 application of knowledge and understanding of computing concepts, application software and the impact of computing

AO3 practical application of skills in using a range of software to produce computational solutions.

Candidates will demonstrate understanding of computing and networking concepts, application software and the impact of computing. They will use relevant application software to produce computational solutions in the form of documents, spreadsheets and charts, as well as demonstrate computational thinking through analysing and debugging programs. Candidates will also apply their skills to create computer graphics, videos and games.

SCHEME OF ASSESSMENT

All candidates will offer Paper 1 and Paper 2. All questions are compulsory in both papers.

Paper 1 (e-Examination, 1 hour 15 minutes, 60 marks)

This paper will assess candidates' knowledge, understanding and application of concepts and skills in all seven modules:

Module 1:	Computer fundamentals
Module 2:	Networking
Module 3:	Impact of computing
Module 4:	Document processing
Module 5:	Spreadsheets
Module 6:	Media software
Module 7:	Programming

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 marks. This paper covers assessment objectives AO1 and AO2.

The paper carries 40% of the total marks and is marked out of 60 marks.

Paper 2 (Lab-based Examination, 2 hours, 90 marks)

This paper, taken with a computer, will assess topics from the following modules:

Module 4:	Document processing
Module 5:	Spreadsheets
Module 6:	Media software
Module 7:	Programming

Candidates will submit softcopies of the required work for marking. The allotted time includes time for saving the required work in the candidates' computer. This paper covers assessment objective AO3.

This paper carries 60% of the total marks and is marked out of 90 marks.

Summary of details for each paper:

Paper	Mode	Duration	Weighting	Marks	Format	Modules Assessed
1	e-Exam	1h 15m	40%	60	Section A 20 Multiple-choice questions [20 marks] Section B Variable number of short structured questions [40 marks]	All modules
2	Lab-based	2h	60%	90	3 Tasks Media software [~30 marks] Document processing and Spreadsheets [~35 marks] Programming [~25 marks]	Modules 4–7

SPECIFICATION TABLE

Assessment Objectives		Paper 1	Paper 2	Overall
AO1	Knowledge and understanding	~20%	–	**~20%**
AO2	Application	~20%	–	**~20%**
AO3	Practical application of skills	–	60%	**60%**
TOTAL		**40%**	**60%**	**100%**

USE OF CALCULATOR

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

CENTRE INFRASTRUCTURE FOR LAB-BASED EXAMINATION

The centre will ensure adequate hardware and software facilities to support the examination of its candidates for Paper 2, which will be administered over at most two shifts on the day of the examination. Each candidate should have the sole use of a laptop for the purpose of the examination. The candidates should be able to access media, document processing, spreadsheet and programming software. The centre must ensure that the approved software is installed in all the candidates' laptops.

SYLLABUS CONTENT

The syllabus consists of seven modules as follows:

Module 1	Computing fundamentals
Module 2	Networking
Module 3	Impact of computing
Module 4	Document processing
Module 5	Spreadsheets
Module 6	Media software
Module 7	Programming

The learning outcomes for each module are shown as follows:

Module 1: Computing fundamentals

1.1: Components

1.1.1 Define computer hardware.
1.1.2 Name the key components of a computer system: its processor, main memory and secondary storage.
1.1.3 State the difference between volatile and non-volatile memory.
1.1.4 State an example of volatile (main memory) and non-volatile memory (secondary storage) respectively.
1.1.5 Give examples of secondary storage media: memory cards, flash drives, hard disk drives and solid state drives.
1.1.6 Compare the sizes of data units: bits, bytes, kilobytes, megabytes, gigabytes, terabytes and petabytes.
1.1.7 Compare computers in terms of processor speed, memory capacity and secondary storage capacity.

1.2: Input and output

1.2.1 Identify the input, process and output of a computer application.
1.2.2 Understand that meaningful information is output only after a computer has processed the correct input data.
1.2.3 Give examples of common input devices: keyboards, mice, touchpads, scanners, barcode readers, digital cameras/webcams and microphones.
1.2.4 Give examples of common output devices: display screens/monitors, printers, speakers/headphones and projectors.

1.3: Software

1.3.1 Define computer software.
1.3.2 Describe the functions of operating systems: to provide users with a user interface, to support the running of application software, to manage the computer's resources and to manage the computer's input and output devices.
1.3.3 Give examples and describe the functions of common software: word processors, spreadsheet software, presentation software, drawing software and programming software.
1.3.4 Distinguish between operating systems and application software.
1.3.5 Give examples of common features of graphical user interfaces: windows, icons, menus and pointers.
1.3.6 State the benefits of file compression: reducing file size and making transfer easier by combining multiple files into a single file.

Module 2: Networking

2.1: Concepts

2.1.1 Understand that computers in a network can facilitate communication and sharing of documents, hardware and software.
2.1.2 Give examples and state the purposes of common computer network devices: network interface cards, wireless routers and modems.
2.1.3 Understand the difference between wired and wireless communications.
2.1.4 Differentiate between local area networks (LANs) and wide area networks (WANs) based on geographical scope.
2.1.5 Understand the difference between intranets and the internet.
2.1.6 Understand the difference between clients and servers in a client-server network.

2.2: Cloud computing

2.2.1 Understand that the cloud refers to computing resources (storage and applications) that are accessed over the internet.
2.2.2 Compare cloud storage to local storage in terms of where files are located and their relative advantages or disadvantages.
2.2.3 Compare cloud-based applications to local applications in terms of relative advantages or disadvantages.
2.2.4 Create custom web-based forms for data collection.
2.2.5 Create simple websites using templates.

Module 3: Impact of computing

3.1: Technology

3.1.1 Give examples of the impact of computers in:

- Communication: ability to connect people and businesses over long distances
- Education: easy access to online classes and large amounts of information via the internet
- Transportation: widespread access to navigational services via Global Positioning System (GPS) and emergence of self-driving vehicles
- Retail: more reliable tracking of available stock and emergence of self-checkout counters

3.2: Responsible use of computers

3.2.1 Understand the online risks associated with scams and malware.
3.2.2 Give examples of malware: viruses, worms, trojans, spyware and ransomware.
3.2.3 Understand how to take measures to prevent falling victim to online risks: use strong passwords, use firewalls, use updated anti-malware programs and identify scam attempts.
3.2.4 Understand how to prevent data loss by making backups for possible recovery in case the originals are damaged.
3.2.5 Understand how to use copyrighted materials responsibly.
3.2.6 Understand the privacy policy and settings of websites before deciding whether to disclose personal information.

Module 4: Document processing

4.1: Body text

4.1.1 Use the find and replace feature.
4.1.2 Paste text with or without formatting.
4.1.3 Insert soft breaks to start new lines without starting new paragraphs.
4.1.4 Set the left, right, first-line and/or hanging indents of a paragraph to specified values.
4.1.5 Use pre-set tab stops to align different lines of text (left-align only).

4.2: Page properties

4.2.1 Give examples of scenarios where single-column and multiple-column layouts may be used.
4.2.2 Create single-column and multiple-column layouts using the built-in columns feature.
4.2.3 Insert headers and footers.
4.2.4 Insert page numbers and set the starting page number to a specified value.
4.2.5 Insert or remove page breaks.
4.2.6 Use section breaks to format a document, including allowing page layout in both portrait and landscape orientation.

4.3: Graphics and text boxes

4.3.1 Insert graphics into a document.
4.3.2 Move, resize, flip, rotate and crop imported graphics.
4.3.3 Embed charts created using spreadsheet software.
4.3.4 Create textboxes and import text into them.
4.3.5 Link text boxes so that text flows from one to another in a specified order.
4.3.6 Modify the borders of a text box.
4.3.7 Set the wrapping style of a text box or graphic.

Module 5: Spreadsheets

5.1: Cell formats

5.1.1 Set cells to use either a number, currency or percentage format with a specified number of decimal places.
5.1.2 Set cells to use a specified date format.
5.1.3 Wrap and align text in cells vertically and horizontally.
5.1.4 Use conditional formatting to change the fill and/or font colour of cells based on their contents. [Limited to “greater than”, “less than” and “equal to”]

5.2: Charts

5.2.1 State the purpose of different chart types: bar charts, column charts, pie charts and line charts.
5.2.2 Create bar charts, column charts, pie charts or line charts with data from either a contiguous or non-contiguous range of cells.
5.2.3 Customise chart elements: chart title, data labels, axes, axis titles and legend.
5.2.4 Recognise that modifying a chart’s data table will cause a corresponding change to the chart.
5.2.5 State the purpose of combination charts.
5.2.6 Create combination charts. [Limited to combination of line and column charts]

5.3: Formulas

5.3.1 Use mathematical operators (+, -, * and /) in formulas.
5.3.2 Use relational operators (>, >=, <, <=, = and <>) to compare values in formulas.
5.3.3 Recognise that the value of cells which use formulas will be automatically recalculated when their referenced cells are changed.
5.3.4 Change the view of a spreadsheet to display formulas.
5.3.5 Use absolute and relative cell referencing.

5.4: Functions

5.4.1 Use a logical function to select between two values based on a logical condition [IF, no nested IFs]
5.4.2 Use mathematical and statistical functions to:

- Round numbers [ROUND]
- Sum numbers [SUM]
- Average numbers [AVERAGE]
- Calculate median, minimum or maximum of numbers [MEDIAN, MIN, MAX]
- Count values (numbers only, non-blank only, with condition) [COUNT, COUNTA, COUNTIF]

5.4.3 Use text functions to:
- Extract characters from text [LEFT, MID, RIGHT]
- Calculate the length of text [LEN]

5.4.4 Use a look up function to look up values from a range of cells using exact matching [VLOOKUP]

5.5: Sorting and filtering

5.5.1 Sort cells in ascending or descending order based on the contents of a particular column.
5.5.2 Filter data by setting criteria on a column. [Limited to “greater than”, “greater than or equal to”, “less than”, “less than or equal to”, “equal to” and “not equal to”]

5.6: Data validation

5.6.1 State why input data may need to be validated.
5.6.2 Set validation criteria for cells.
5.6.3 Display custom error messages when invalid input data is keyed in.

Module 6: Media software

6.1: Media elements

6.1.1 State that text, graphics, animation, sound and video are media elements.
6.1.2 State that multimedia content consists of multiple media elements.
6.1.3 Identify the different types of media elements used in some specified multimedia content.

6.2: Vector graphics

6.2.1 Explain that vector graphics are created using nodes and paths.
6.2.2 State that vector graphics can be resized without loss of quality.
6.2.3 Create drawings using lines, curves, text, ellipses and polygons.
6.2.4 Move, resize, rotate and flip objects.
6.2.5 Duplicate/copy and delete objects.
6.2.6 Set the front to back arrangement of objects.
6.2.7 Group multiple objects into a single object and ungroup them again.
6.2.8 Recognise solid fills, gradient fills and pattern fills.
6.2.9 Set the fill of objects using a specified colour and style.
6.2.10 Set the transparency of objects such that objects underneath them are visible.
6.2.11 Set the outline of objects using a specified colour and thickness.
6.2.12 Put text to follow a curved path.
6.2.13 Export vector graphics as raster graphics.
6.2.14 Create complex shapes using merging features: union, intersect, fragment, subtract, combine.
6.2.15 Modify objects by manipulating their nodes and node handles directly.

6.3: Raster graphics

6.3.1 State and recognise that raster graphics are composed of individually coloured pixels.
6.3.2 Give PNG, GIF, TIFF, BMP and JPEG as examples of different file formats for raster graphics and state if transparency is supported for each file format.
6.3.3 State that resizing raster graphics can result in a loss of quality.
6.3.4 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.
6.3.5 Adjust the sharpness, brightness, contrast and dimensions (resize and crop) of raster graphics.

6.4: Presentations and videos

6.4.1 Insert media elements into a slide presentation.
6.4.2 Adjust the relative dimensions of media elements and use white space appropriately.
6.4.3 Use appropriate fonts and colours to enhance readability.
6.4.4 Use appropriate font styles and sizes to distinguish between headings and body text.
6.4.5 Use appropriate backgrounds for a slide presentation to enhance contrast with media elements.
6.4.6 Use the slide master feature to achieve a consistent style and layout.
6.4.7 Produce slide presentations based on storyboards.
6.4.8 Understand that frames are individual images in a video.
6.4.9 Create a video from still images and videos with text, transitions and sound.
6.4.10 State that videos with higher frame rates may take up more space but may also appear smoother than videos with lower frame rates.

Module 7: Programming

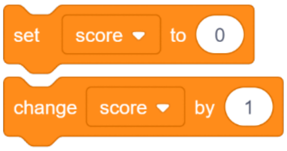
7.1: Basics

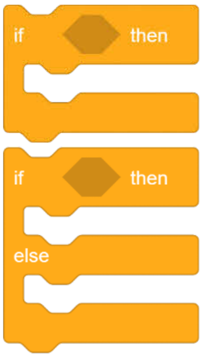
7.1.1 Recognise that visual programming languages use graphical symbols to develop programs.
7.1.2 State that visual programming can be used to create games.
7.1.3 Represent the sequence of events in a proposed game using words, sketches and/or storyboards.
7.1.4 Interpret flowcharts to understand a program's sequence of events.
7.1.5 State that the purpose of variables is to store values.
7.1.6 Create and name variables.
7.1.7 Initialise and update the values of variables.
7.1.8 Use conditional instructions (if and if-else). [No nested ifs]
7.1.9 Use basic loops (repeat, forever).
7.1.10 Use conditional loops (repeat-until).
7.1.11 Use logical (or, and, not) and relational operators (>, < and =) in conditional instructions and/or loops.
7.1.12 Generate and use random numbers in programs.
7.1.13 Use mathematical operators (+, -, * and /) in programs.
7.1.14 Identify and correct errors in programs.

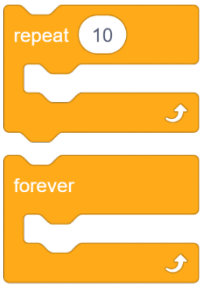
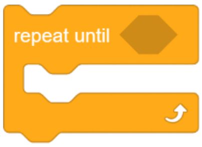
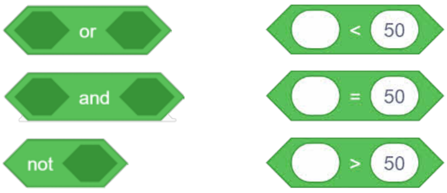
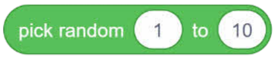
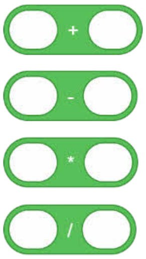
7.2: Game programming

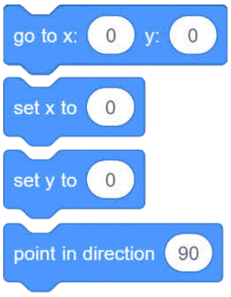
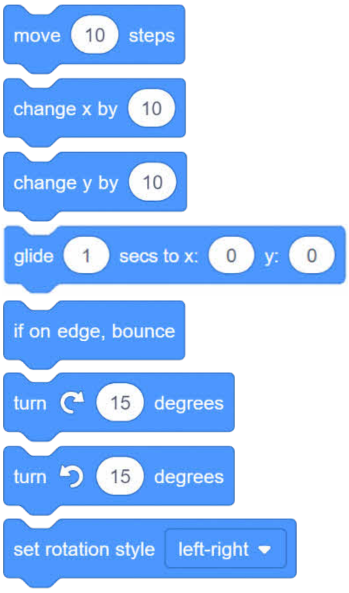
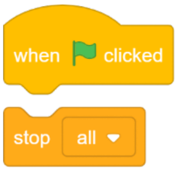
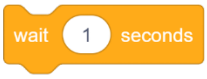
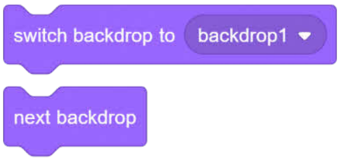
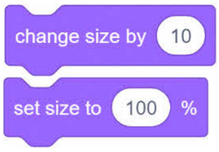
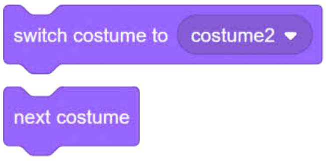
7.2.1 Recognise that points on the stage can be represented using their x and y coordinates.
7.2.2 State that code consists of instructions to be executed by a sprite or the stage.
7.2.3 Recognise that multiple sets of code can be executed at the same time.
7.2.4 Position sprites at a specified location and orientation.
7.2.5 Move and rotate sprites.
7.2.6 Start and stop the execution of code.
7.2.7 Insert wait time between the execution of two instructions.
7.2.8 Insert additional backdrops to the stage by choosing from the library, importing from a file or drawing with the built-in editor.
7.2.9 Switch between the stage’s backdrops.
7.2.10 Create and name sprites.
7.2.11 Change the size of sprites.
7.2.12 Insert additional costumes for a sprite by choosing from the library, importing from a file or drawing with the built-in editor.
7.2.13 Switch between a sprite’s costumes.
7.2.14 Show and hide sprites.
7.2.15 Display text as either a speech or thought bubble.
7.2.16 Play sounds for an object.
7.2.17 Understand how to record and store digital voice.
7.2.18 Understand how to edit sound clips by performing trim, insert and volume control operations.
7.2.19 Display and hide the values of variables.
7.2.20 Prompt for and accept text input.
7.2.21 Send a message to trigger other objects to start executing their code.
7.2.22 Set key presses and/or mouse clicks to trigger execution of code.
7.2.23 Use contact between sprites and/or coloured areas of objects in conditional instructions and/or loops.

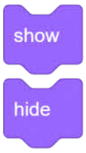
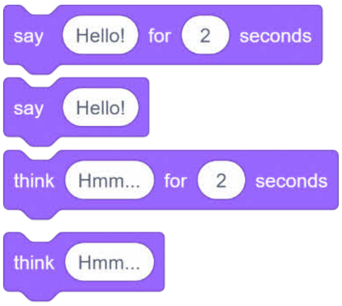
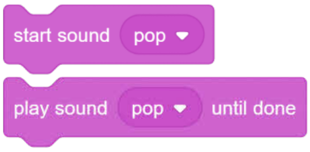
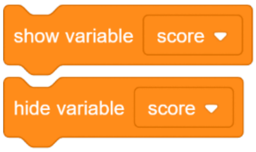
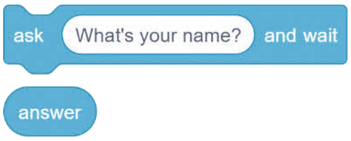
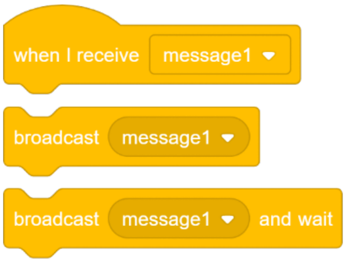
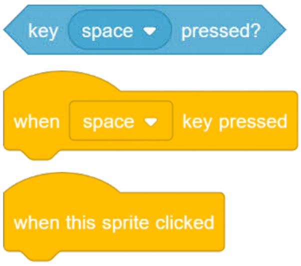
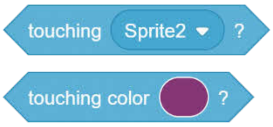
List of Examinable Scratch Blocks

Learning Outcomes 7.1: Basics	**Scratch Blocks**
7.1.7 Initialise and update the values of variables.	 set score ▾ to 0 change score ▾ by 1

7.1.8 Use conditional instructions (if and if-else). [No nested ifs]	
	if then
	if then
	else

7.1.9 Use basic loops (repeat, forever).	 repeat 10 forever
7.1.10 Use conditional loops (repeat-until).	 repeat until
7.1.11 Use logical (or, and, not) and relational operators (>, < and =) in conditional instructions and/or loops.	 or and not < 50 = 50 > 50
7.1.12 Generate and use random numbers in programs.	 pick random 1 to 10
7.1.13 Use mathematical operators (+, -, * and /) in programs.	 + - * /

Learning Outcomes 7.2: Game programming	**Scratch Blocks**
7.2.4 Position sprites at a specified location and orientation.	 go to x: 0 y: 0 set x to 0 set y to 0 point in direction 90
7.2.5 Move and rotate sprites.	 move 10 steps change x by 10 change y by 10 glide 1 secs to x: 0 y: 0 if on edge, bounce turn ↻ 15 degrees turn ↺ 15 degrees set rotation style left-right ▾
7.2.6 Start and stop the execution of code.	 when ⚑ clicked stop all ▾ (**Note:** The stop block is not always required to stop execution of scripts)
7.2.7 Insert wait time between the executions of two instructions.	 wait 1 seconds
7.2.9 Switch between the stage's backdrops.	 switch backdrop to backdrop1 ▾ next backdrop
7.2.11 Change the size of sprites.	 change size by 10 set size to 100 %
7.2.13 Switch between a sprite's costumes.	 switch costume to costume2 ▾ next costume

Learning Outcomes 7.2: Game programming	**Scratch Blocks**
7.2.14 Show and hide sprites.	 show hide
7.2.15 Display text as either a speech or thought bubble.	 say Hello! for 2 seconds say Hello! think Hmm... for 2 seconds think Hmm...
7.2.16 Play sounds for an object.	 start sound pop ▾ play sound pop ▾ until done
7.2.19 Display and hide the values of variables.	 show variable score ▾ hide variable score ▾
7.2.20 Prompt for and accept text input.	 ask What's your name? and wait answer
7.2.21 Send a message to trigger other objects to start executing their code.	 when I receive message1 ▾ broadcast message1 ▾ broadcast message1 ▾ and wait
7.2.22 Set key presses and/or mouse clicks to trigger execution of code.	 key space ▾ pressed? when space ▾ key pressed when this sprite clicked
7.2.23 Use contact between sprites and/or coloured areas of objects in conditional instructions and/or loops.	 touching Sprite2 ▾ ? touching color ● ?