2224–6912M

Markscheme

May 2024

Computer science

Standard level

Paper 1

15 pages

Subject details: Computer science SL paper 1 markscheme

Mark allocation

Section A: Candidates are required to answer **all** questions. Total 25 marks.
Section B: Candidates are required to answer **all** questions. Total 45 marks.
Maximum total = 70 marks.

General

A markscheme often has more specific points worthy of a mark than the total allows. This is intentional. Do not award more than the maximum marks allowed for that part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

- Each statement worth one point has a separate line and the end is signified by means of a semi-colon (;).
- An alternative answer or wording is indicated in the markscheme by a "/"; either wording can be accepted.
- Words in (…) in the markscheme are not necessary to gain the mark.
- If the candidate's answer has the same meaning or can be clearly interpreted as being the same as that in the markscheme then award the mark.
- Mark positively. Give candidates credit for what they have achieved and for what they have got correct, rather than penalizing them for what they have not achieved or what they have got wrong.
- Remember that many candidates are writing in a second language; be forgiving of minor linguistic slips. In this subject effective communication is more important than grammatical accuracy.
- Occasionally, a part of a question may require a calculation whose answer is required for subsequent parts. If an error is made in the first part then it should be penalized. However, if the incorrect answer is used correctly in subsequent parts then **follow through** marks should be awarded. Indicate this with "**FT**".

General guidance

Issue	Guidance
Answering more than the quantity of responses prescribed in the questions	• In the case of an "identify" question, read all answers and mark positively up to the maximum marks. Disregard incorrect answers. • In the case of a "describe" question, which asks for a certain number of facts *eg* "describe two kinds", mark the first two correct answers. This could include two descriptions, one description and one identification, or two identifications. • In the case of an "explain" question, which asks for a specified number of explanations *eg* "explain two reasons …", mark the first two correct answers. This could include two full explanations, one explanation, one partial explanation *etc*.

Section A

1. (a) *Award* ***[1 max]***

Easily deployable and can be implemented on demand;
Cost effective - flexible payment methods such as pay - as- you -go models;
Can be accessed anywhere / anytime / from any device;
Reduction in costs as no need to host software installation / less hardware / pay personnel / rental costs;
Scalability / can increase subscription or reduce it as required very quickly;
Software always / regularly up to date at the supplier's side without the need for installing patches / updates / upgrades;
Ensured compatibility / easier collaboration - all users will have the same version of the software;
Increased security;
Easier administration - less maintenance / less hardware required;
Technical support given by the provider;

(b) *Award* ***[1 max]***

Confidentiality concerns if data related to the customer is also stored on the supplier's hardware;
Stability issues / latency issues - dependence on good quality internet connection can mean if the internet is down / low quality connection, no work / slow work;
If the service goes bankrupt / shuts down you can lose your applications and data;
Critical business functions are left in the control of a third party / Organization has no direct control of its own data;
If the quality of service provided by the supplier is not up to the standard, data loss or corruption can occur;
Limited customization;
Issues with interoperability and data exchange between different platforms due to incompatible / limited API.
Security concerns as higher risk of failure / attack / interception;
Time Delay in contacting / receiving response from the supplier / customer support if there is an issue / different time zone;
Can turn out to be costly in the long run;

2. *Award* ***[2 max]***
Award ***[1]*** *for the description and award* ***[1]*** *for the extension of its purpose*

Examples:

One area/branch of the business is selected to try the new computer system;
Once the new system is fully tested and able to function correctly / without errors / as per the user requirements, it is implemented for everyone;

The system is introduced only to a small group of users;
It reduces the risks of failure or malfunction before introducing to the full group;

Pilot running is a small-scale implementation of the system for a limited number of users;
It allows the developers make improvements in the system based on users' feedback before its full release / roll out;

3. *Award **[2 max]***
Award ***[1]*** *for the purpose and award* ***[1]*** *for an appropriate extension*

Examples:

User acceptance testing evaluates a product/software to make sure it is up to specifications/ user requirements/functions as intended;
before putting it into production/ before release;

It tests for the functionality of the product;
ensuring that bug fixes / modifications can be completed (in a more cost-effective manner);

It serves as a form of quality check;
to identify problems and defects through user feedback;

4. *Award **[2 max]***

On a CD / DVD / Blu-ray disc / flash drive;
(In built) Help files;
Website / Online support (in the form of FAQs / chat sessions) / Video demonstrations;
Printed manuals / PDFs;

Note to examiners: accept answers relating to methods of supplying or media used to present user documentation. Do not accept answers related to training.

5. *Award **[2 max]***
*Award **[1]** for the method and award **[1]** for correct description*

Self-instruction / Self-study / Self-learning;
The user would work through documentation / watch instructional videos / learn the features of the application / system by themselves and understand how to use it;

Formal classes / Instructor led training / Face to face training;
Organised (online / classroom) sessions in which a tutor would train a class of users in specific features of an application or system;

Online training / Distance training
Users are trained over the net / over a digital platform / remotely using training materials designed to show them how to use the application or system;

Note: Accept other suitable responses (Workshops / Seminars / On the job training) with suitable description.

6. *Award* ***[1 max]***

A hardware/device that is <u>connected to / attached to / external / auxiliary to the computer.</u>
A device used for input, output or storage that is <u>external / attached to a computer.</u>

7. *Award* ***[2 max]***

Very small text and readability;
Very small functional elements / difficult to navigate;
Clutter on the screen;
Low contrast between text and functional elements / poor colour choices for visibility;
Size of screen (difficult to see in poor light);
Size of keys (difficult to access functions);
Battery life (may need to recharge frequently);
Touch screen keys (lack of tactile feedback);
Gesture (unresponsive gestures in user interfaces);
Too many clicks to complete tasks;
Limited storage / RAM;

Note: Reward other suitable issues

8. *Award* ***[2 max]***

Use of touch screen;
Voice recognition;
Braille keyboards;
Text-to-speech/Screen readers;
Hotkeys;
Larger screens;
Larger text options;
Size and sensitivity of touchpad – larger / less sensitive pads;

Note: Reward other suitable answers ;

9. (a) *Award* ***[2 max]***

To hold the data / instructions;
to be transferred/fetched to / from the main memory / RAM of the computer OR that has been read or needs to be written;
using the address from the MAR;

(b) *Award* ***[2 max]***

The ALU performs simple arithmetic operations / mathematical calculations / addition, subtraction, multiplication, division;
… and logical comparisons / AND, OR and NOT operations;

10. (a) *Award* ***[1 max]***
00100101 *OR* 100101 *OR* 0100101;

(b) *Award* ***[1 max]***
9E;

11. *Award* ***[2 max]***
Award ***[1]*** *for each comparison up to* ***[2]***

RAM is a volatile / temporary memory (which could store the data as long as the power is supplied) and ROM is a permanent / non-volatile memory (which could retain the data even when power is turned off);
Data stored in RAM can be altered whilst data stored in ROM can only be read;
RAM is used to store the instructions / data / programs that are currently being processed / in use whereas ROM is used to store permanent data / files such as start-up instructions for the computer.
RAM - large physical chip size / higher capacity / expensive whilst ROM - small size / less capacity / cheaper;
The CPU can access the data stored in RAM directly whilst the CPU cannot access the data stored on ROM (unless the data is stored in RAM) / whilst ROM data can be accessed after transferring it to RAM

Note: Do not award mark for Cache memory.

12. *Award* ***[2 max]***
Physical Layer;
Data Link Layer;
Network (layer);
Transport Layer;
Session Layer;
Presentation Layer;
Application Layer;

Section B

13. (a)(i) *Award* ***[2 max]***

Allows very fast transmission of data / Extremely high bandwidth possible;
(Made of glass / plastic fibre) that can transfer information via pulses of light;
Immune to electromagnetic interference / temperature changes / severe weather / highly resistant to noise and moisture;
Very high security rating;
Safe to use in high-voltage locations, areas where flammable gases / chemicals;
Very long distance of transmission before requiring repeaters / attenuation;
They have a long lifespan (thinner and light weighted, so more flexible than other media);
Expensive / harder to install;

(ii) *Award* ***[2 max]***

Uses radio waves / electromagnetic waves to transfer data;
Transmission speed of data is limited;
Range of transmission / transmission reliability can be affected by distance from access point / number of other users/ obstacles etc;
Inexpensive to install/ no need to spend on cabling;
Relatively easy to expand/ add new devices / scale down to accommodate changes in demand;
Security can be poor (unless encryption is applied) / subject to eavesdropping / interception;
Allows users to move around without losing access to the network;

(b) *Award* ***[3 max]***

Plain text is changed to cypher text / Data is scrambled using an encryption algorithm / key / A Key is required by sender and receiver for authentication;
Cypher text / data cannot be understood if intercepted;
The cypher text/data is then decrypted using a (decryption) key when received by the receiver;

(c) *Award* ***[5 max]***

The whole data is split into (fixed / equal) sized chunks / packets;
Each packet has a header, payload and trailer;
Packet contains information such as source, destination IP addresses, packet number, protocol, checksum, payload / data, CRC etc (at least two);
Each packet is sent individually / independently along the best path (by a router);
Packets may take different routes to the destination;
If a route becomes unavailable, individual packets can be re-routed;
Packets (may) arrive at the destination out of order;
Packets are re-ordered / joined together at the destination;
Missing packets can be re-sent;

(d) *Award **[3 max]***

*Award **[1]** mark for implication (positive or negative) and award **[2]** for further expansion*

Example 1
Improved work / life balance;
Working from home / access to work materials has become much easier / more secure which can allow more time to be spent at home;
Time spent on travelling to work can now be spent with the family / on hobbies and interests / travelling;
Also, working hours can be flexible, so the employee is less likely to feel stressed leading to improved mental health;

Example 2
It can have a negative impact on work / life balance;
With working from home some workers may feel that their employer has higher expectations of the amount of work they should be doing because of the time they have saved not having to travel to work;
Also, some workers may feel that their employer doesn’t trust that they are fully committed to working when at home / distracted by family, so feel under pressure to spend more time on the job/cannot concentrate enough on job, leading to poorer mental health;

Example 3:
Working from home may also result in less collaboration / team work.
As working at different times / zones may not allow employees to communicate effectively with each other;
and feeling of isolation can affect mental health and productivity;

Note: No marks if there is no social implication mentioned.

14. (a) *Award* ***[1 max]***
Award ***[1]*** *for either the description OR the truth table.*

Outputs the value of one if and only if all inputs have a value of zero;

A	**B**	**Z**
0	0	1
0	1	0
1	0	0
1	1	0

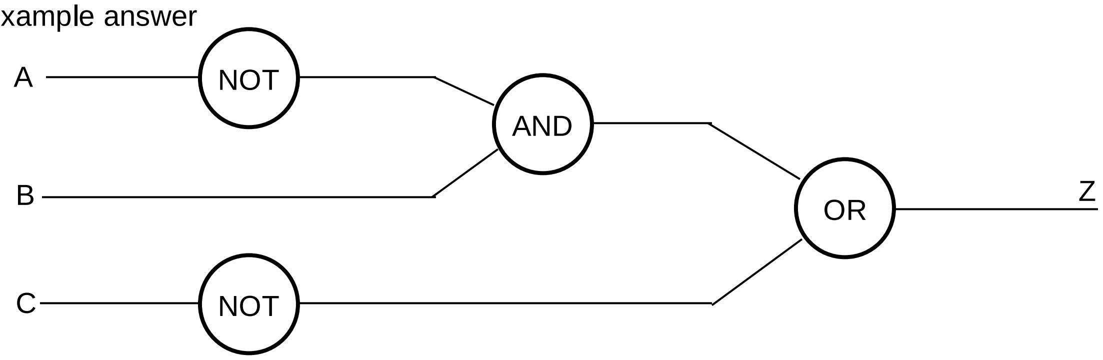
Note - DO NOT accept - reverse / negates OR

(b) *Award* ***[4 max]***

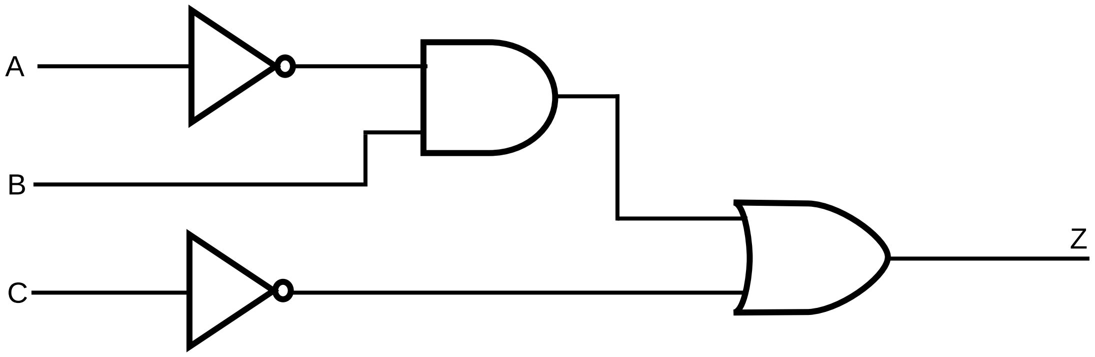
Correct NOT gate with input A and single output;
Correct NOT gate with input C and single output;
Correct AND gate with two inputs - one input from B and one output;
Correct OR gate with two inputs and one output Z;

Answer should represent Z = A’.B + C’

Example answer



Alternative answer



(c) *Award **[2 max]***
*Award **[1]** for each correct row (The rules for 0 and 1 can be reversed).*

D	0	Obstruction is less than 3 metres from the rear of the car / There are obstruction less than 3 metres from the rear of the car
	1	Obstruction is 3 or more metres from the rear of the car / There are no obstructions less than 3 metres from the rear of the car

(d) *Award **[2 max]***
*Award **[1]** for evidence of working e.g. substitution into variables to evaluate the expression;*
*Award **[1]** for correct answer false;*

Example answer

```
5 >= 25 AND 30 >= 10 AND 30 <= 130 AND true
false AND true AND true AND true
```

(Output =) `false`

Note: Accept 1/0 instead of TRUE/FALSE

(e) *Award **[6 max]***

Use of flag (or otherwise) to maintain continuous loop;
Appropriate loop structure – while / repeat until;
Inputs for distance and speed inside the loop;
Correct conditions to implement alarm notification;
Correct Output of alarm;
Correct condition to cause loop to stop;

***Note**: Do not accept break to terminate the loop*

Example answer 1

```
FLAG = true
loop while FLAG
    input D // allow read D
    input S // allow read S
    if D < 20 OR S > 130 then
        output "ALARM!!"// allow any other output
    end if
    if S = 0 then
        FLAG = false
    end if
end loop
```

*Award **[6 max]***

Input of speed before the loop;
Appropriate loop structure – while / repeat until;
Inputs for distance and speed- after the endif. Both inside the loop;
Correct condition to implement alarm notification;
Correct Output of alarm;
Correct condition to cause loop to stop;

Example answer 2

```
input S
loop while S > 0
    input D // allow read D
    if D < 20 OR S > 130 then
        output "ALARM!!"
    end if
    input S // allow read S
end loop
```

15. (a) *Award **[2 max]***

addItem();
adds an element to the collection (at the end);

getNext();
returns / gets / obtains the next element from the collection (indicated by the pointer);

resetNext();
restarts the iteration through the collection from the first element / moves the pointer to the beginning of the collection;

hasNext();
returns TRUE if there is as least one more element in the collection that has not been accessed (in the current iteration) / if there is any element remaining to access;

isEmpty();
returns TRUE if the collection does not contain any elements;

*Note: Award **[1]** mark for name and Award **[1]** mark for the description of collection methods.
Accept other equivalent method names such as add(), getData() etc.*

Alternative mark points based on syllabus content:
*Award **[1 max**];*

Data can be added to the collection (at the current location or at the start);
Data cannot be inserted at a specific location in the collection;
Data can be removed/retrieved from the collection (at the current location or at the start);

(b) *Award* ***[1 max]***

Correct use of collection methods `hasNext(),getNext() (, resetNext())` or equivalent methods;

Award ***[3 max]***
Initializing the starting array index as 0;
Appropriate loop for accessing RAINBOW elements;
Correct assignment to both arrays using the same index, with two adjacent elements from RAINBOW;
Array indexes increment for whole data set;

Example answer

```
RAINBOW.resetNext()
I = 0
loop while RAINBOW.hasNext()
   COLOUR[I] = RAINBOW.getNext()
   ORDER[I] = RAINBOW.getNext()
   I = I + 1
end loop
```

(c) *Award* ***[5 max]***

Bubble Sort
Use of nested loops with correct conditions / initial and final values;
Use of flag to stop the loop if no swaps made;
Comparing adjacent elements of `ORDER` array;
Swapping contents in `ORDER` array if required;
Corresponding swapping of contents in `COLOUR` array if required;
Note: Do not accept break statement.

Example 1

```
SWAP = TRUE
loop while SWAP = TRUE
  SWAP = FALSE
  loop I from 0 to 5
    if ORDER[I] > ORDER[I + 1] then
      TEMP1 = ORDER[I]
      TEMP2 = COLOUR[I]
      ORDER[I] = ORDER[I + 1]
      COLOUR[I] = COLOUR[I + 1]
      ORDER[I + 1] = TEMP1
      COLOUR[I + 1] = TEMP2
      SWAP = TRUE
    end if
  end loop
end loop
```

Bubble Sort
Use of nested loops;
Looping with the correct initial and final values;
Comparing adjacent elements of `ORDER` array;
Swapping contents in `ORDER` array if required;
Corresponding swapping of contents in `COLOUR` array if required;

Example 2

```
loop J from 0 to 5 // Accept loop J from 0 to 6
   loop I from 0 to 6-J-1
      if ORDER[I] > ORDER[I + 1] then
            TEMP1 = ORDER[I]
            TEMP2 = COLOUR[I]
            ORDER[I] = ORDER[I + 1]
            COLOUR[I] = COLOUR[I + 1]
            ORDER[I + 1] = TEMP1
            COLOUR[I + 1] = TEMP2
      end if
   end loop
end loop
```

Selection Sort
Use of nested loops with correct initial and final values;
Setting of `MIN` using correct index before the inner loop;
Comparing elements in `ORDER` array to find smallest value inside the correct loop;
Swapping contents in `ORDER` array inside the correct loop;
Corresponding swapping of contents in `COLOUR` array;

Example 3

```
loop I from 0 to 5 // Accept loop I from 0 to 6
  MIN = I
  loop J from I + 1 to 6
    if ORDER[J] < ORDER[MIN] then
      MIN = J
    end if
  end loop
  if min <> I then
    TEMP1 = ORDER[MIN]
    TEMP2 = COLOUR[MIN]
    ORDER[MIN] = ORDER[I]
    COLOUR[MIN] = COLOUR[I]
    ORDER[I] = TEMP1
    COLOUR[I] = TEMP2
  end if
end loop
```

Selection Sort
Use of nested loops with correct initial and final values;
Setting of `MIN` value and correct initialization of `POS`;
Comparing elements in `ORDER` array to find smallest value and storing the `POS`;
Swapping contents in `ORDER` array inside the correct loop;
Corresponding swapping of contents in `COLOUR` array;

Example 4

```
loop I from 0 to 5 // Accept loop I from 0 to 6
  MIN = ORDER[I}
  POS = I
  loop J from I + 1 to 6
    if ORDER[J] < MIN then
      MIN = ORDER[J}
      POS = J
    end if
  end loop
  if POS <> I then
    TEMP1 = ORDER[MIN]
    TEMP2 = COLOUR[MIN]
    ORDER[MIN] = ORDER[I]
    COLOUR[MIN] = COLOUR[I]
    ORDER[I] = TEMP1
    COLOUR[I] = TEMP2
  end if
end loop
```

(d) *Award* ***[4 max]***
Note: At least one similarity and one difference to be mentioned

Similarities:
Both use nested loops, (each time reducing the inner loop);
Both have the same worst case time complexity ($O(n^2)$);
Both perform in place sorting directly within the array (without requiring additional data structures / memory);

Differences:
Bubble sort works by comparing adjacent elements, swapping the elements if they are in wrong order, (repeating these steps until the array is fully sorted);
Selection sort works by selecting the smallest (or largest) element from the unsorted portion of the array and swapping it with the first unsorted element;
Bubble sort can exit early/ is faster if the list is already sorted;
Selection sort will need to complete the procedure for the entire list every time;
Bubble sort has a higher number of comparisons and swaps, making it slower compared to selection sort for larger datasets;
Bubble sort has a best-case time complexity O(n) whereas Selection sort has a best-case time complexity $O(n^2)$;