Markscheme

May 2016

Computer science

Standard level

Paper 1

General marking instructions

1. Follow the markscheme provided, award only whole marks and mark only in **RED**.
2. Make sure that the question you are about to mark is highlighted in the mark panel on the right-hand side of the screen.
3. Where a mark is awarded, a tick/check (✓) **must** be placed in the text at the **precise point** where it becomes clear that the candidate deserves the mark. **One tick to be shown for each mark awarded.**
4. Sometimes, careful consideration is required to decide whether or not to award a mark. In these cases use RM™ Assessor annotations to support your decision. You are encouraged to write comments where it helps clarity, especially for re-marking purposes. Use a text box for these additional comments. It should be remembered that the script may be returned to the candidate.
5. Personal codes/notations are unacceptable.
6. Where an answer to a part question is worth no marks but the candidate has attempted the part question, enter a zero in the mark panel on the right-hand side of the screen. Where an answer to a part question is worth no marks because the candidate has not attempted the part question, enter an "NR" in the mark panel on the right-hand side of the screen.
7. Ensure that you have viewed **every** page including any additional sheets. Please ensure that you stamp 'SEEN' on any page that contains no other annotation.
8. A mark should not be awarded where there is contradiction within an answer. Make a comment to this effect using a text box or the "CON" stamp.

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. *Award up to [2 max].*
Award [1] for evidence that the candidate knows what is meant by "maintaining legacy systems".
Award [1] for any appropriate problem outlined.

Example:
Maintaining previous/outdated computer system, which uses old technology and
old application programs ;
That are hard to understand/expensive to change/evolve **because** programs might be disorganized/
documentation might be missing/incomplete/unreliable;
Compatibility issues (typically refer to old programming languages/old database technology);
Maintaining but not updating the old system can lead to compatibility / security issues;
It may be difficult to recruit staff/programmers familiar with old languages/operating systems;
Database contains inconsistencies/redundancies (eg information systems and no DBMS);
Usually pre-internet, needs interfaces;
Typically large and complex systems/mainframes;   **[2]**

2. A system or a partially functional prototype;
Given to users to test for functionality or to gain feedback on functions or the user interface;   **[2]**

3. *Award [1] for **one** advantage and [1] for **one** disadvantage, and [1] for expanding on*
each of them, up to [4 max].

Examples of advantages:
* Portability;
  Printed material is more easily transportable and can be moved around (*eg* for a
  scanner/printer);
* Extent of material;
  Books/technical instructions for installation may have more details, and be more
  useful to provide deeper explanations;
* Availability;
  It is always available (no power cut problems);

Examples of disadvantages:
* Readability;
  Font size online can be (easily) magnified;
* Trouble-shooting/cross-reference/usability;
  Online is faster and usually has links to other pages whereas paper is a thick manual;
* Aging;
  Online is more frequently updated than paper manuals;
* Environmental;
  Waste of paper versus energy consumption;   **[4]**

Note: *Accept expansions of these or other plausible advantages/disadvantages.*

4. *Award up to [2 max].*
A failover system is a standby/redundant system;
Which is used to eliminate/reduce the impact on users/owners;
By automatically taking over if the primary system suddenly becomes unavailable;   **[2]**

5.   *Award up to [2 max].*
Obtains the data/instructions from the memory;
Interprets/decodes them into commands/steps/signals;
Controls transfer of data and instructions among other units of a CPU (for example, command to ALU for execution);
Manages/coordinates all the units of the computer;
etc.   **[2]**

6.   *Award up to [2 max].*
Cache memory is closer to CPU/faster to access than main memory/incorporated on the chip;
By holding recently/frequently used data and instructions in cache;
Execution of program/fetching instructions and data is faster;   **[2]**

7.   *Award [1] for one feature and [1] for a reason/explanation.*
Examples:
Memory management;
Game applications use a lot of memory and require constant refreshing;

Processor loading of OS functions (efficiency);
Graphics handling of OS (as distinct from graphics card),

GUI;
O/S needs to handle input from appropriate devices;   **[2]**

8.   *Award [3] marks for all 8 correct rows in the truth table.*
Award [2] marks if only 6/7 correct rows in the truth table.
Award [1] mark if only 5 correct rows in the truth table.

A	B	C	A XOR (B OR C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

                                                                                                  [3]

9.   *Award [1] for correct 111011.*
Award [1] for using two leading zeroes for the 8-bit register.

00111011;   **[2]**

10.   *Award **[1]** for correct output.*
*Award **[1]** for each of the **three** correct iterations (three rows).*

D	Z	B	Z<L ?	output
46	1	false	true	
15	2	true	true	
5	3	false	false	
				(3 , true)

[4]

Section B

11. (a) *Award up to **[2 max]** for **two** aspects:*

The type of access needed;
For example read only/read write/online or offline;

Access rights;
For example, data available only for administrators / different permissions for students;

Frequency of access;
Some data (of non-current students) are not frequently accessed and can be archived;
Other data (of current students) are frequently used, subject to a variety of operations;

Quantity/size of the data;
For example should not exceed storage capacity of the new system;

Type/nature/format;
For example incompatibility issues;   **[2]**

(b) *Award up to **[3 max]**.*
Quick/first hand/realistic information on data/software/hardware/users/procedures in the current system;
Help better understand positive and negative features of the current system (for example problems in accessing or validating data/user errors/security issues, etc.);
Which can be used when specifying requirements of the new system (keep/improve positive and change negative features);   **[3]**

(c) *Award up to **[3 max]**.*
Prototype is used to ensure all essential functions/operations of the system are present/meets the needs of the users;
Prototype is used to speed up development process;
Positive user's feedback helps in refining the acceptable prototype in order to develop the complete system/product;
Or else a further prototype should be created in order to develop the satisfactory system/product;
etc.   **[3]**

(d)   *Award [1] for identifying a problem and [1] for a discussion, for **two problems**, up to [4 max].*

Data loss/data corruption;
When moving data, from one storage device to another (via network/ cables or transferred by people), data could be corrupted/lost and not useful anymore;

Incompatibility of data formats;
Necessary to translate from one format to another, to be able to use the data in the new system which causes delays/performance issues in business/office operation;   **[4]**

(e)   *Award up to [3 max].*

Example answer 1
Two systems are running simultaneously so that operations are not disrupted;
This is a costly operation;
Because both systems and all their resources should be maintained / More staff should be hired;

Example answer 2
Safe way of validating the new system;
Running two systems could be cheaper;
Than losing all data in case of failure;   **[3]**

12. (a) *Award [2] for an explanation of the reliability of wireless.*
Award [2] for an explanation of the reliability of Ethernet
Award up to [4 max].
Note: *Do not accept answers relating to security.*

WIRELESS
The reliability of wireless depends
  on the strength of the wireless signal/distance from router;
  on the topology/shape of the surroundings;
  on interference/number of simultaneous connections on an access point;

ETHERNET
Ethernet is more reliable as the strength of the signal is independent from the
distance from the router (within the college);
There is no issue with the topology/shape of the surrounding, as long as the user has a
connection;
connection depends on condition of cables – no loose or broken cable connections;   **[4]**

(b) *Award [1] for identifying the feature, up to **two** features.*
*Award [1] for an expansion of **two** features.*
Award up to [4 max].

Authentication;
Nobody outside the VPN should be able to affect the security property of the VPN
(it must be impossible for the attacker to weaken/change encryption);

Encryption;
Data intercepted will not be readable;

Tunnelling software;
Security properties of each tunnel should be agreed by the administrators of the
two endpoints of the tunnel;

Multiple exit nodes;
Makes it hard to distinguish where the data was generated thus more secure (less
prone to phishing);   **[4]**

(c) *Award [1 max].*
SSL 3.0 (Secure socket layer 3);
TLS (with encryption) (Transport Layer Security);
IPsec with encryption;   **[1]**
Note: *Do not accept "internet connection".*

(d)   *Award **[1]** for **each** advantage, up to **[2 max]**, **[1]** for **each** disadvantage, up to **[2 max]**, and **[2]** for a discussion in relation to an example.*

Answers may include (and are not limited to) the following elements:

Advantages:
May offer more recent technology than the college itself;
The college cannot replicate the social aspects of diffused discussion/social networking;
Registration/creation/access of account is usually easy/cheap/free;
Allows interaction with others/collaborative studies/exchange of materials/opinions;
It means students can access and coordinate data and assignments on any device and from any location;
Note: *Do not accept "can reach work when not at college" on its own, as there is a VPN for this purpose.*

Disadvantages:
One might post things they would regret later on;
The content is potentially available across the world;
The content is available for a long time;
The content submitted to external services might not be fully in line with the internal policy, even if posting it is allowed (offensive content);
Photos/documents may be hacked by third parties;

For example:
Posting offensive comments;
Ownership / confidentiality / security of data;
Takedown policy from external providers (if the posted content is offensive, the service provider may take a while to take it down);   **[6]**

13. (a)   150 (= 450/3);   **[1]**

(b)   73;   **[1]**
*Note: Accept `PARTICIPANTS[73]`. Do not accept `TEAM[11]` because the
question asks specifically for the array `PARTICIPANTS`.*

(c)   (i)   *Award marks as follows up to [3 max].*
Award [1] for looping through PARTICIPANT.
Award [1] for checking PARTICIPANT [P] = CURRENT.
Award [1] for exit when MINP found – accept 'break'.
Award [1] for storing the index in MINP.

Example 1:
```
loop while PARTICIPANTS[P] ≠ CURRENT AND P<450
  P = P+1
endloop
MINP = P
T = MINP
```

Example 2:

```
boolean FOUND = FALSE
loop while FOUND = FALSE and P < 450
  if PARTICIPANTS[P] = CURRENT then
    FOUND = TRUE
    MINP = P
  else
     P = P + 1
  end if
end loop
```

Example 3:

```
boolean FOUND = FALSE
loop P from 0 to 450-1
  if PARTICIPANTS[P] = CURRENT then
    FOUND = TRUE
    MINP = P
    BREAK
  end if
end loop
```

                                                                                         [3]

(ii) *Award marks as follows up to **[4 max]**.*
*Award **[1]** for a correct loop from P to 449 (either incremented in (c)(i) or stated here).*
*Award **[1]** for starting at T=MINP*
*Award **[1]** for updating T index.*
*Award **[1]** for correct value of P in TEAM[T].*
*Award **[1]** for closing the path by assigning MINP.*

Example 1:
```
T = MINP // can be here or in c(i)
loop while P<450
   if PARTICIPANTS[P] = CURRENT then
     TEAM[T] = P
     T = P
   end if
   P = P+1
end loop
TEAM[T] = MINP
```

Example 2:
```
loop while P<450
   if PARTICIPANTS[P] ≠ CURRENT then
     P = P+1
   else
     TEAM[T] = P
     T = P
     P = P+1
   end if
end loop
TEAM[T] = MINP
```
[4]

(d)   *Award up to **[6 max]** for covering the following points.*

Describe and use new data structures/variables;
Loop through TNAMES for each team;
Retrieve/access TIMINGS for the times of each member from TEAM;
(Store in array or list)
Pass team member times to sum3best;
Store result of sum3best for team in array or variable;
Identify winning team;

Example 1
Create array TEAMTIMES with same length as TNAMES
Create array TEMPTIMES with length 5 (max number of participants per team)
For each team in TNAMES
    For each team member in TEAM find the corresponding time in TIMING
      Insert the time into the next element of TEMPTIMES (assumption: times over 120 min
      have already been assigned a time of 1000)
    End loop
    Pass TEMPTIMES to function sum3best
    Insert result of sum3best into TEAMTIMES at same position as current element of TNAMES
End loop
Find smallest value in TEAMTIMES
Set TEAM = TNAMES[*index of smallest value in* TEAMTIMES]

Example 2
Create array TTIMES max 5 to hold times in one team
Create FASTEST to hold time of winning team
Create BEST to hold name of winning team
Set FASTEST = 5000 (default time for 5 team members or something)
For each team in TNAMES
    use PARTICIPANTS and TEAM to find index for each member in TIMINGS
    store in TTIMES array of max 5
    Pass array TTIMES to sum3best and store in SUM
    If SUM < FASTEST set FASTEST to SUM and set BEST to TNAME
End loop
winning team = BEST   **[6]**