M16/4/COMSC/HP1/ENG/TZ0/XX/M

Markscheme

May 2016

Computer science

Higher level

Paper 1

20 pages

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 HL 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 75 marks.
Maximum total = 100 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. *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]**

3. *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]**

4. *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]

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

00111011; **[2]**

6. 3, 4, 5, 7, 1, 2, 9, 11; **[1]**
***Note:** Some responses might apply the bubble sort to the list, from right-to-left. The answer in this case would be* 1, 2, 9, 11, 3, 4, 5, 7, and this should receive full credit.

7. *Award up to **[3 max]**.*

Virtual machines provide a software emulation/virtualization of other operating systems;
While hiding/abstracting the physical environment/OS/execution environment of the host machine;
Virtual machines do not directly correspond to any real platform so software may be tested/executed on any platform (portability);
This is more practical/economical than having to test software on several computers with different OS;
The software to be tested is compiled into the language of the virtual machine, and this is interpreted into the language of the host machine; **[3]**

8. *Award up to **[3 max]**.*
*Award **[2]** for description of VoIP.*
*Award **[1]** for an explanation related to one (or more than one resource).*
***Note:** Award **[1]** only if resources are mentioned without any explanation.*

Example:
Audio (and video) stream transmitted over internet;
Broadband internet connection needed;

That can be integrated in an office desktop computer;
with collaboration desk endpoints/webcams/desktop computer/widgets/
touch screen/specific applications/IP phones;

Examples:

Bandwidth:
Available Bandwidth / A trade-off between bandwidth and quality of audio-video streaming;
Low bandwidth makes streaming disrupted/discontinuous and impacts on the quality of the collaborative environment;
Bandwidth also depends on how many people in the office will use VoIP (and possibly the network in general) and their simultaneous operations (uploading/downloading);

Router:
Router should be powerful/good enough to support all operations related to collaborative work;
Some specific routers provide quality of service for voice transmission over other functions, for example WiFi/VPN connections;
Normal routers do not distinguish the internet traffic, and the quality of audio-video streaming may decrease;

Network:
Quality of internet varies, for example delays/congestion are cause of losses/echoes/scattered images;
Having two separated networks for VoIP and other computer operations may help in some cases (depending on the dimensions and operations of the office), so that the router can handle them separately;
For example somebody talking over the IP is not hindered by somebody else in the same office downloading from the internet;

Power:
In case of power loss the telephones will also not work;
A backup system/second system just for the telephones is necessary; **[3]**

9. *Award **[2]** for each of the two functions, up to **[4 max]**.*

Memory management;
OS allows more than one program/process to share the memory;
By allocating separate memory to each program;
Provides memory isolation for each of processes;
The system may begin to run out of shared storage (as many programs are running) so OS moves pages to disk/paging;
etc.

Processor Management;
To allow (the appearance of) more than one program running at the same time;
By the allocation of time slices;
Decides which process runs at a certain point in time;
Arrange the execution of applications so that you believe that there are several things happening at once (scheduler);
Prioritizes tasks by importance (interrupts);
etc. **[4]**

10. *Award **[1]** for a rationale, **[1]** for an example of misuse, and **[1]** for a way to prevent it, up to **[3 max]**.*

The rationale:
The right of the company to protect their premises/assets/IP shall not invade privacy rights of the employee;

Therefore precise guidelines must be set to prevent misuse of technology from the employer and ensure rights to the employee;

Examples of Misuse:
CCTV might not be used for surveillance only, but also to monitor employees;

It is a threat to personal privacy through intimidation/harassment/checking absentees/screening with consequences on quality of life/wellbeing/employment;

For example: monitoring employee's activities/look for negative behaviour/gestures / focusing on some groups of employees (women/LGBT/ethnic or faith groups, *etc*);

For example: different cultures may interpret gestures in different ways, and this might be used to reprimand an employee;

Moving equipment (also a USB stick with data) from one building to another is monitored, but the footage can be interpreted as theft;

Prevent Misuse:
A secondary system for labelling equipment should be used (tagging items), to avoid misinterpretations, and discharge incorrect incriminations (but it is more subtle with IP);

Problem of where the footage is stored (company premises), for how long, and who has access to the information, for what use, shall also be addressed;

Surveillance staff (*ie* those who have access to CCTV footage) shall not be employees of the company or there could be internal conflicting situations (*ie* an employee controlling another employee); **[3]**

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) *Award **[1]** for a feature of a linked list.*
*Award **[1]** for relating it to the given scenario.*

Size of a linked list is not fixed/predetermined (efficient use of memory);
Suitable because the number of objects/cars in the car park may vary greatly;

Efficient addition of elements to the linked list;
Suitable because cars arrive at the car park in any order; **[2]**

(b) (i) *Award **[1 max]**.*
Raising both barriers;
Payment of the ticket at the machine, with any other process; **[1]**

(ii) *Award **[1 max]**.*
Incrementing and decrementing the display cannot happen concurrently;
Updating the linked list/collection following a ticket issue and a payment;
The barrier raising and the ticket being checked;
etc. **[1]**

(c) *Award **[1 max]**.*
The ticket has been paid at most 10 minutes ago;
Accept formulations such as `CurrentTime – PaidAt <= 10`; **[1]**

(d) *Award up to **[3 max]**.*
*Award **[1]** for evidence that operations on exit & display are **before** the operations at entrance & display.*
*Award **[1]** for **correct sequence** of operations at the exit.*
*Award **[1]** for **correct sequence** of operations at the entrance.*

At the exit barrier
(Check ticket)→raise exit barrier→increment display;

Then at the entry barrier
(Issue ticket) →decrement display→ raise entry barrier; **[3]**

(e) *Award up to **[2 max]**.*

Example answer (space)
Eventually the collection is too large and it could run out of memory;
This impacts on the issuing of further tickets;

Example answer (time)
Sequential access and the growing linked list means that accessing the node takes longer;
This impact on the time needed for issuing/checking tickets (cars are waiting longer in the queue to leave/enter the car park); **[2]**

(f) *Award marks as follows up to **[5 max]**.*

*Award **[1]** for calculating excess dates (XD) (as a step only – no precise code required).*
*Award **[1]** for a correct calculation of payment due for fines lasting more than one day, upon test.*
*Award **[1]** calculating parking duration (DRN) within the same day (as a step only, no precise code required).*
*Award **[1]** for correct calculation of payment due for parking duration lasting less than 2 hours (within the same day) (CASE 1 and CASE 2).*
*Award **[1]** for correct calculation of payment due for parking duration lasting more than 2 and less than 4 hours (within the same day) (CASE 3 and CASE 4).*
*Award **[1]** for correct calculation of payment due for parking duration lasting more than 4 hours (within the same day).*

For example:

Compute excess dates (XD), using arrival date (AD) and current date (CD)
(*eg* XD = CD–AD);

If XD ≠ 0 then PAY= XD*RULES[1][2] (accept PAY = XD*30)
(*ie* if the dates are not the same, the amount to pay is the maximum fine for XD days);

Else compute parking duration (DRN), using arrival time (AT) and current time (CT) (*eg* DRN = CT–AT);

After suitable transformation into minutes;

```
Case (1):  DRN =< 30         then PAY = RULES [0][0]
           (accept 0.50 EUR)
Case (2):  30<DRN=<120       then PAY = RULES [0][1]
           (accept 3.00 EUR)
Case (3):  120<DRN=<150      then PAY = RULES [0][1] + RULES [1][0]
           (accept 3+5 = 8 EUR)
Case (4):  150<DRN=<240      then PAY= RULES [0][1] + RULES [1][1]
           (accept 3+15= 18 EUR)
Case (5):  240<DRN           then PAY = RULES [1][2]
           (accept 30 EUR)
```

[5]

Notes: *Other solutions are possible, award marks accordingly.*
It is not necessary to refer to RULES (and relative notation) as a double array.
Accept constructions with flowcharts or pseudocode.
Accept calculations that make use of div and mod, provided they are correct and respect the specification of a daily-based fee system.
Accept DRN presented in hours.
Award marks for logic, not for the syntax.

Example answer:

```
if dateofissue IS NOT equal to dateofpayment then
  calculate number of days(XD)
  PAY=XD*30     //more than 1 day
else
  calculate parking duration in minutes (DRN)
  if DRN>240 then
    PAY=30      //more than 4h within one day
  else if DRN>150 then
    PAY=18  //more than 2.5h less than 4h
  else if DRN>120 then
    PAY=8       //more than 2h less than 2.5h
  else if DRN>30 then
    PAY=3       //less than 2h and more than 0.5h
  else PAY=0.5  //less than 0.5
  end if
end if
```

14. (a) 6; **[1]**

(b) Award *up to* ***[2 max]*** *for computing the user's location.*
Award up to ***[2 max]*** *for drawing the zones on the map.*

The GPS receiver in the Smartphone picks up the signals from 3 satellites (at least 3);
The signals transmitted are: time of transmission, coordinates of the satellite;
The receiver knows when the signal was received;
Calculate positioning though equation resolution on a sphere;

Zones are displayed relative to the user's current position;
And the current scale of the map;
For each distance required, a circle is drawn (on the map) to define the zone, centred on the current GPS position; **[4]**

(c) *Award marks as follows up to **[5 max]**.*
*Award **[1]** for initialization of two variables (for most likes (ML) and the name of restaurant with most likes (BR)).*
*Award **[1]** for correct loop.*
*Award **[1]** for testing whether the item in the collection is in Zone 1.*
*Award **[1]** for jumping elements (or not using) data items which are not in Zone 1.*
*Award **[1]** for comparing current likes with most likes (ML) and **correct** update of ML (if needed).*
*Award **[1]** to output **correct** result (award this mark only if BR is updated each time ML is changed).*

***Note:** Award marks for the logic, not the syntax.*

Example answer 1

```
ML = 0
BR = ""          // Initialization
loop while LIKES.hasNext()
  Z = LIKES.getnext()
  L = LIKES.getnext()
  N = LIKES.getnext()
  X = LIKES.getnext()
    // Instantiate Z,L,N,X with elements from LIKES
    // that relate to the same restaurant
  if Z==1 and L > ML then
    ML=L
    BR=N
  End if    // Compare current with most likes and update ML and BR
end loop
output(BR) // The name of restaurant with most likes is output
```

Example answer 2

```
ML = 0
BR = ""          // Initialization
LIKES.resetnext();  // Go to start of the collection
A = LIKES.getnext();
loop while not LIKES.isempty()
  if A ≠ 1 then
    A = LIKES.getnext().getnext().getnext().getnext();
      // the next element of LIKES referring to the zone
      // is four elements away
  else
    B = LIKES.getnext();  // Instantiate B,C,D with elements
    C = LIKES.getnext();  // from LIKES that relate to the same
    D = LIKES.getnext();  // restaurant in zone 1
    //Compare and update if applicable
    if B>ML then
      ML = B;
      BR = C;
    end if
  end if
end loop
output(BR);// The name of restaurant with most likes is output.
```

Example answer 3
*(**Note:** This model answer uses Java methods; although not part of the core topics, candidates not penalized for presenting answers in Java.)*

```
ML = 0
BR = ""         // Initialization
loop while LIKES.hasNext()
  if LIKES.getNext()==1 then
    if LIKES.getNext()>ML   // Compare and update ML and BR then
      ML=LIKES.getCurrent()
      BR=LIKES.getNext()
      LIKES.skipNext(1) //  Jumping 1 element in collection
    end if
  else
    LIKES.skipNext(3) //  Jumping 3 elements
  end if
end loop
output(BR) // The name of restaurant with most likes is output
```

[5]

(d) *Award marks as follows up to **[3 max]**.*
*Award **[1]** for associating trees to zones.*
*Award **[1]** for explaining how to use the collection's elements to build the trees.*
*Award **[1]** for inserting nodes under some criterion.*
*Award **[1]** for visiting the trees in such a way as to produce the ordered list.*

For example:
Create three binary trees (one for each zone), by taking 4 elements at a time from `LIKES`;
The element "zone" identifies the tree **and** the other three elements make a new node in that tree;
Nodes are added depending on "likes" so that left child =< (accept "<") the (subtree) root, less than right child;
Do an inorder traversal on the three trees to get three lists, then join them together; **[3]**

(e) *Award **[1]** for **one** disadvantage and **[1]** for an elaboration, up to **[2 max]**.*

For example:
Feedback/reviews might not correspond to trustworthy judgement;
And call something cheap that is not cheap / something good that is not so good;
The same person might use more than one identity to add weight to opinions;
There is a market for fake reviews, and people may even be paid to author them;
It is hard to counterbalance negative and not trustworthy feedback for the business because of the complexity of the review system; **[2]**

***Note:** Accept other plausible answers.*

15. (a) 21; **[1]**

(b) *Award marks as follows up to* ***[3 max]****.*
Award ***[1]*** *for evidence of* ***two recursive calls*** *of* `fibo(2)` *or of two recursive calls of* `fibo(0)`*.*
Award ***[1]*** *for evidence of* ***three recursive calls*** *of* `fibo(1)`*.*
Award ***[1]*** *for correctly returning values* `0` *and* `1` *for* `fibo(0)` *and* `fibo(1)` *when needed.*

For example:

```
fibo(4)
= (fibo(3)+fibo(2))
= (fibo(2)+fibo(1)) + (fibo(1)+fibo(0))
= ((fibo(1)+fibo(0)) +1) + (1+0)
= ((1+0)+1) + (1+0)
= 3
```

[3]

Note: *No marks shall be given if the answer 3 is provided without any tracing, nor if the intermediate values instead of calls to* `fibo()` *are given; this is a tracing question.*
The order of evaluation of intermediate steps may be different from the one presented in the solution here.

(c) *Award marks as follows, up to* ***[6 max]****.*
Award ***[1]*** *for initialization.*
Award ***[1]*** *for correct condition in If statement.*
Award ***[1]*** *for correct loop within if statement.*
Award ***[1]*** *for correctly summing up.*
Award ***[1]*** *for correct swap of variables in the loop.*
Award ***[1]*** *for correct return/output.*

Example 1:

```
fibo(N)
if N==0 OR N=1
  RES = N     // RES stores the result
else
  BACK2 = 0 // Initialize variables
  BACK1 = 1
  loop J from 2 to N
    RES = BACK1 + BACK2
    BACK2 = BACK1
    BACK1 = RES
  end loop
end if
return RES
```

Example 2:

```
fibo(N)
V = 1    // Initialize variables
RES = 1  // RES stores the result
if (N=0) then
  return N
else
  loop J from 3 to N
    TEMP = V + RES
    V = RES
    RES = TEMP
  end loop
end if
return RES
```

[6]

(d) *Award marks as follows, up to* ***[3 max]****.*
Award ***[1]*** *for setting up a loop.*
Award ***[1]*** *for calling* `fibo()`*.*
Award ***[1]*** *for outputting the result.*

Example 1:

```
procedure(N)
loop J from 0 to N
  output(fibo(J))
end loop
```

Example 2:

```
procedure(N);
arrayfibo[N]   // declare array with N positions
loop J from 0 to N
  arrayfibo[J] = fibo(J)
end loop
H = 0
loop while H=<N
  output arrayfibo[H];
    H = H+1;
end loop
```

[3]

Note: *Accept comparable solutions, also in flowcharts, provided they are correct.*

(e) The current environment (*eg* values/local variables/current address/registers) PUSHED onto the stack when a new recursive call is met;

To be POPPED OFF the stack when the recursive subprogram is completed. **[2]**