M15/4/COMSC/HP1/ENG/TZ0/XX/M

Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

May 2015

Computer science

Higher level

Paper 1

13 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 marks max]****.*
Award ***[1 mark]*** *for communication with user – email/ pop up etc.*
Award ***[1 mark]*** *for method of installation of update – automatic/ link/ in list for user to install etc.*

When the software is installed and registered (a cookie is placed on the machine);
This communicates with the software developer automatically on start up;
Messages about updates are sent back to the machine and alerts are given;

OR

Send an email;
With a link to the update; **[2]**

2. *Award* ***[1 mark]*** *for each correct pair of rows.*

A	B	C	(A or B) and (not C or B)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

[4]

3. *Award* ***[1 mark]*** *for a relevant example and* ***[2 marks]*** *for an elaboration.*

Example 1:
A business can let employees work at home / employees who travel a lot/external (non-employee) users;
Accessing the data and services (at the office);
Via secure login;

Example 2:
Using VPN, address is masked;
The location of the user is not known;
May be essential in delicate situations such as political protest groups working from their own country;
Note: *Accept any legitimate reason for needing to be unknown.* **[3]**

4. *Award* ***[4 marks]*** *as follows.*
Award ***[1 mark]*** *for going 3 times through the loop (with COUNT from 1 to 3).*
Award ***[1 mark]*** *for incrementing correctly SUM (when N mod COUNT = 0).*
Award ***[1 mark]*** *for the correct output ("perfect").*
Award ***[1 mark]*** *for showing all working in a trace table with at least three columns (eg COUNT, SUM, OUTPUT).*
Award ***the first 3*** *marks for an evident trace but working not shown in a trace table.*

Example answer 1:

COUNT	N mod COUNT=0	SUM	SUM=N	output
1	TRUE	1		
2	TRUE	3		
3	TRUE	6		
			TRUE	perfect

Example answer 2:

COUNT	N mod COUNT	SUM	output
		0	
1	0	1	
2	0	3	
3	0	6	perfect

[4]

5. *Award marks as follows up to* ***[3 marks max]****.*
Award ***[2 marks max] for the first three passes*** *correct,*
*(****[1 mark]*** *for at least one of passes 1, 2 and 3 correct).*
Award ***[1 mark]*** *for correct passes 4 and 5 with no change on pass 4.*

Pass	12	52	16	42	88	86
1	**88**	**52**	**16**	**42**	**12**	**86**
2	**88**	**86**	**16**	**42**	**12**	**52**
3	**88**	**86**	**52**	**42**	**12**	**16**
4	**88**	**86**	**52**	**42**	**12**	**16**
5	**88**	**86**	**52**	**42**	**16**	**12**

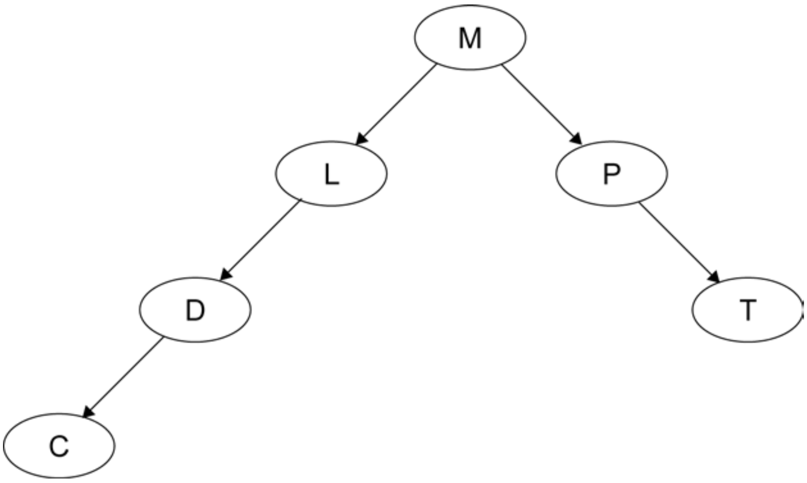
[3]

6. *Award up to* ***[3 marks max]****.*
GPS works by communication with satellites;
By knowing the position of the satellite (sent to GPS device);
And calculating the time difference between satellites;
The position of the device can be calculated; **[3]**

7. *Award* ***[1 mark]*** *for an appropriate use for the user,* ***[2 marks]*** *for an elaboration.*
For example:
Icon showing images on the user's desktop;
Connects to (part of) the server/printer;
Operating system runs this access in the background (device drivers); **[3]**

8. (a) CDLHTPM **[1]**

(b)



[2]

Section B

9. (a) *Award marks as follows,* ***up to [6 marks max]****.*
Award ***[1 mark]*** *for looping through the database and accessing all records.*
Award ***[1 mark]*** *for correct calculation of date difference (eg = today – paymentDate* ***OR*** *paymentDate – today).*
Award ***[1 mark]*** *for each list correctly compiled,* ***x3*** *(correct conditional statements according to date difference used).*
Note: *Accept date difference not calculated/stated but assumed as today – paymentDate* ***OR*** *paymentDate – today.*
Award ***[1 mark]*** *for successive if/else but wrong conditions.*
Award ***[1 mark]*** *for flagging correct records for deletion (do not accept deleting the records).*

Example:

```
set CURRDATE to current date (as a day number)
set LIST1, LIST2 and LIST3 to empty
loop through all CUSTREC in DATABASE
     DUEPERIOD = CURRDATE - CUSTREC.PAYMENTDATE
     if DUEPERIOD > 30 then
          add CUSTREC to LIST3
          flag CUSTREC to delete
     else if DUEPERIOD > 14 then
          add CUSTREC to LIST2
     else if  DUEPERIOD < - 30 then
          add CUSTREC to LIST1
     end if
end loop
```

[6]

Note: *If candidates give their answer in flowchart form then credit them using the same marking points.*

(b) *Award up to* ***[4 marks max]****.*
(Using a mail merge facility);
Template for each type of reminder created in the word processor;
Lists created with customer ID;
Linked to customer details in database;
Appropriate details merged/inserted into template; **[4]**

(c) *Award* ***[1 mark]*** *for a consequence of data loss to customers and* ***[1 mark]*** *for a consequence of data loss to the insurance company.*

Example answer:
Customers would not be reminded when they needed to pay and some may overlook payment, hence not be insured;
The company could lose customers/ruin reputation; **[2]**

(d) *Award marks as follows up to **[3 marks max]**.*
*Award **[1 mark]** for a suitable measure and **[2 marks]** for a description related to the insurance company.*

Example answers:
Mirror system;
All changes to the records made on two systems;
If one fails then the other holds all current data;

Off site backup;
Snapshots/backups made on a regular basis;
In the case of failure a dated/time stamped copy exists and the state up until then can be used to restore customer records; **[3]**

10. (a) The OSI is a standardized system/model for network connection;
Consists of (7) layers;
Each dealing with specific parts of network communication;
For example, the physical layer which defines the physical connection; **[3]**

***Note:** Award **[1 mark]** for the purpose of any of the 7 layers.*
*If candidate lists all 7 layers with no specific example award **[2 marks]** and a further **[1 mark]** if the purpose of at least one layer is given.*

(b) *Award up to **[3 marks max]**.*
Protocols are a set of rules;
To facilitate a process being carried out correctly;
(Used in each layer to ensure communication;)
For example (in the physical layer) the protocols could define the methods for opening and closing communication; **[3]**
***Note:** Do not accept examples which are not related to networks.*

(c) *Award up to **[2 marks max]**.*
Name/ID;
Whether or not they are already a client;
If not a client, further details needed as input; **[2]**

(d) Two dimensional array;
With one column for each lawyer;
And one row per time slot; **[3]**
***Note:** Accept column or row for lawyer and vice versa for time slot.*

(e) *Award up to **[5 marks max]**. Accept answers given as an algorithm.*
Loading ‘today’ page (and from now onwards or accept “start with tomorrow”);
If existing client, search appropriate lawyer/column only;
If not existing client, search time/row then lawyer/column;
Then allocate space if available;
If no space allocated, load next page and repeat until space found;
Then add client details to space; **[5]**

11. (a) *Award* ***[1 mark]*** *for data,* ***[1 mark]*** *for pointers,* ***[1 mark]*** *for order.*

Example:

Each node would hold the data for one plane (ID, place, time due, time expected, landed);
Head pointer points to the first in the list;
Each subsequent pointer points to the next in the list and last node has null pointer; **[3]**

(b) *Award* ***[1 mark]*** *for calculating hours.*
Award ***[1 mark]*** *for calculating minutes.*
Award ***[1 mark]*** *for input and output/return.*

Example 1:

```
input CTIME // time held in the collection in minutes
     HOURS = CTIME div 60
     MINUTES = CTIME mod 60
output HOURS, MINUTES // time to be displayed on the screen
```

Example 2:

```
input CTIME // time held in the collection in minutes
HOURS = 0
MINUTES = CTIME
WHILE MINUTES>59
     MINUTES=MINUTES-60
     HOURS=HOURS+1
ENDWHILE
output HOURS, MINUTES // time to be displayed on the screen
```

Example 3:

```
Format24 (CTIME)
// method accepts time held in the collection in minutes
     HOURS = CTIME div 60
     MINUTES = CTIME mod 60
     return  HOURS + “:” + MINUTES
     // returns time to be displayed on the screen
end Format24
```

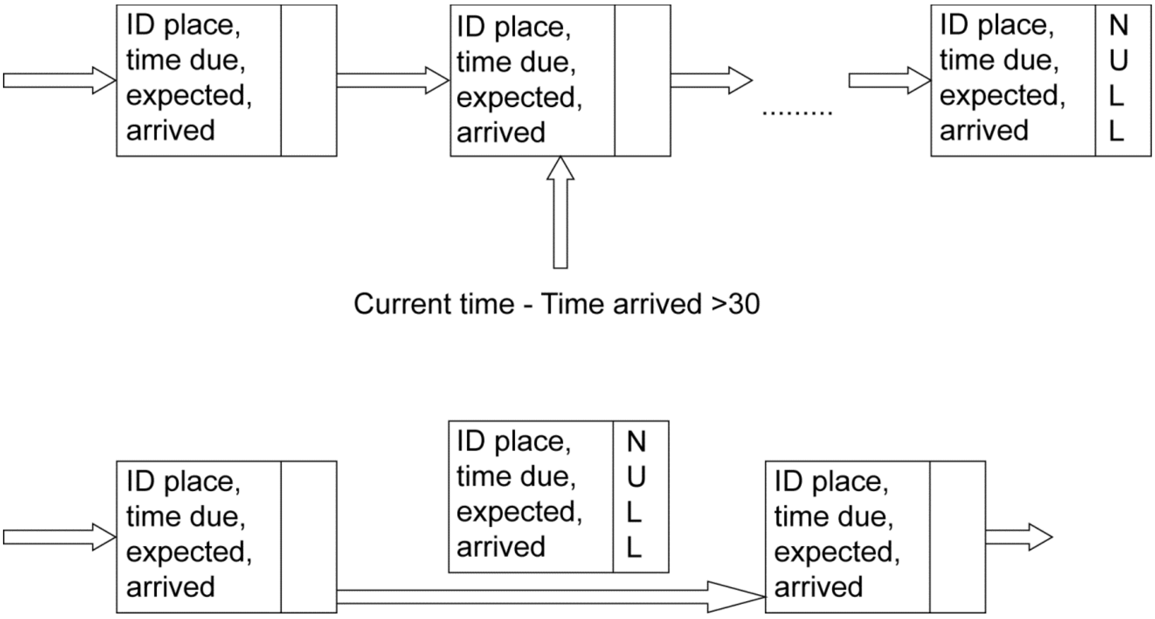
[3]

(c) *Award marks as follows, up to* ***[4 marks max]****.*
Award ***[1 mark]*** *for a diagram and explanation showing access to each plane via pointers;*
Award ***[1 mark]*** *for comparison of current time with time arrived;*
Award ***[1 mark]*** *for correct change of pointer from plane deleted;*
Award ***[1 mark]*** *for correct change of pointer to next plane;*

Note: *The plane to be deleted could be at the beginning of the list* ***OR*** *at the end of the list* ***OR*** *in the middle of the list; award third and fourth mark (change of pointers) depending on the position of the node shown in the candidates' diagram/explanation.*

For example:

```
PLANES accessed sequentially via pointers;
PLANE.ARRIVED checked against current time;
    if > 30 minutes;
        if pointer is head pointer;
            move head pointer to point to next PLANE;
        else if plane is last in list previous pointer points to NULL;
        else previous pointer changed to subsequent plane;
        pointer of deleted plane null;
```


ID place, time due, expected, arrived
ID place, time due, expected, arrived
....
ID place, time due, expected, arrived
NULL
Current time - Time arrived >30
ID place, time due, expected, arrived
ID place, time due, expected, arrived
NULL
ID place, time due, expected, arrived

[4]

(d) *Award up to* ***[5 marks max]****.*
A queue would hold the elements in order of arrival;
And enqueue correctly to the end as required;

Dequeue would take planes from the top of the screen;
Which is not wanted as they arrive at different times;

Elements in a linked list could be removed from any position in the list;
Hence a linked list is better;

Searching for ID to amend will be equivalent; **[5]**

12. (a) Sensors will be used to detect the approach of a vehicle from the minor road;
Likely to be touch/weight sensor embedded in the road;
Sensor input is converted from analog to digital;
To be processed and;
Signal sent to switch traffic lights; **[4]**

(b) *Award up to **[3 marks max]**.*
Continual feedback from sensor to processor;
A calculation based on number of vehicles/speed/*etc* or time taken for a vehicle to pass;
(Timer) resets if another vehicle is detected;
Once no input for a certain time traffic lights changed back; **[3]**

(c) *Award **[2 marks]** for advantages, **[2 marks]** for disadvantages and **[1 mark]** for weighing up.*

Disadvantages:
Central computer would have to cope with inputs from many places;
With differing priorities which could take time;
Connection failure possible from a particular point;
Computer failure puts all lights in the area out;
Cost of communication system/central control system;

Advantages:
More control over traffic flow at these points;
Lights can be adapted from distance to avoid traffic blocks;
Any problem appearing at one point is known immediately and can be dealt with;
Cheaper as no need for communication software/hardware/control centre;
Can react/change rules to changing levels of traffic flow;

Overall, it would be better to ... *(appropriate conclusion)*; **[5]**

(d) *Award **[1 mark]** for an advantage outlined, **[1 mark]** for a disadvantage outlined and **[1 mark]** for discussing.*

For example:
Controlling the movement of vehicles and identifying people who speed should help to reduce accidents (as motorists know that they will be caught if driving dangerously);
This could also save lives;

Individual displacement is tracked;
Which can be seen as an infringement on personal liberty/a breach of privacy;
In some cases the information could be used unjustly against the individual (*eg* in times of political unrest);

It comes down to physical safety on the road against privacy/personal liberty; **[3]**

13. (a) Because items/moves are needed in reverse order;
To the order input;

OR

Because it is a LIFO (Last In First Out) data structure;
The items/moves pushed/placed onto the stack;
Will be popped off/taken from it in reverse order (to the order input); **[2]**

(b) *Award marks as follows, up to **[5 marks max]**.*
*Award **[1 mark]** for checking for empty.*
*Award **[1 mark]** for popping STEP and TURN from stack.*
*Award **[1 mark]** for an output (STEP).*
*Award **[1 mark]** for correct if statement (checking popped TURN).*
*Award **[1 mark]** for an output (TURN).*

For example:

```
(let STK be the stack)
  while NOT STK.isEmpty() do
    STEP = STK.pop()
    output "Take " + STEP + " steps"
    TURN = STK.pop()
    if TURN = 0 then
      output "Turn left"
    else
      output "Turn right"
    end if
  end while
```

[5]

(c) (18, 5) **[1]**

(d) *Award marks as follows up to **[6 marks max]**.*
*Award **[1 mark]** for each correct pair X and Y (coordinates), **x3**.*
*Award **[1 mark]** for each correct change of direction facing (correct E and N), **x3**.*

Move		**Coordinates**		**Direction facing**	
TURN	STEP	X	Y	N	E
		0	0	1	0
0	10	10	0	0	1
1	5	10	5	1	0
0	8	**18**	**5**	**0**	**1**
0	2	**18**	**3**	**-1**	**0**
1	3	**21**	**3**	**0**	**1**
0	0				

[6]