

# **Markscheme**

**May 2017**

**Computer science**






















**Higher level**


**Paper 1**

This markscheme is **confidential** and for the exclusive use of examiners in this examination session.

It is the property of the International Baccalaureate and must **not** be reproduced or distributed to any other person without the authorization of the IB Global Centre, Cardiff.

The following are the annotations available to use when marking responses.

Annotation	Explanation	Associated shortcut
	Good Response/Good Point	(Alt+) 0
	Incorrect Point	(Alt+) 1
	Unclear	
	Benefit of the doubt	
	No benefit of doubt	
	Seen	
	Too vague	
	Repetition	
	Follow through	(Alt+) 2
	(Comp Sci) Language	(Alt+) 3
	Description	(Alt+) 4
	Analysis	(Alt+) 5
	Reference	(Alt+) 6
	Development	(Alt+) 7
	Balanced argument	
	Off course	(Alt+) 8
	Evaluation	
	Opinion	
	Dynamic, horizontal line that can be expanded	(Alt+) 9
	Dynamic, horizontal wavy line that can be expanded	
	Dynamic, vertical wavy line that can be expanded	
	Text box	

You **must** make sure you have looked at all pages. Please put the  annotation on any blank page, to indicate that you have seen it.

### 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	<ul style="list-style-type: none"> <li>• In the case of an “identify” question, read all answers and mark positively up to the maximum marks. Disregard incorrect answers.</li> <li>• 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.</li> <li>• 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.</li> </ul>

## Section A

1. It is when an application/program is tested/ to get feedback (for errors/improvements);  
By releasing it to the general public/users/stakeholders before final release; [2]
  
2. *Award up to [2 max].*  
Natural disaster/power failure;  
Accidental deletion;  
Malicious activities (hacking/theft/viruses);  
Hardware/software/system failure;  
Bad integration/migration of systems/data;  
Transmission error;  
Lossy compression when the original no longer available only the modified version; [2]
  
3. *Award up to [2 max].*  
To patch any vulnerabilities/bugs/cyberspace threats;  
To provide improved functionality/new functions/usability/maximise efficiencies;  
To generate income for the software company/to innovate and stay ahead of other software companies;  
To ensure compatibility with other (updated) technologies; [2]
  
4. Cache memory can be accessed/is faster than RAM;  
It is used to hold common/expected/frequently used data/operations;  
Closer to CPU than RAM/situated between RAM and CPU/on same board as CPU/with faster read/write speed;  
Cache memory is static RAM and this memory doesn't need to be constantly refreshed; [2]
  
5. The OS allocates (and deallocates) specific sections of memory to each program/process/module;  
This ensures that the memory assigned to one program is not overwritten;  
Uses secondary/virtual memory to allow more processes to run simultaneously;  
**Note:** Do not accept vague reasons. [2]
  
6. *Award up to [2 max] for any two of the following:*  
Physical layer;  
Data link layer;  
Network layer;  
Transport layer;  
Session layer;  
Presentation layer;  
Application layer; [2]
  
7. Some compression methods (lossy) discard data;  
Decompression will not return the complete file/some detail will have been removed;  
Which in some cases eg audio/video may be unacceptable;  
If original not saved/lost there is no way to recover it; [3]

8. *Award up to [3 max].*  
The MAC address identifies a specific device (network card/controller);  
MAC address checked against list of approved addresses/whitelist If not on list access to network is denied;  
Prevents unauthorised access/makes access more difficult/(unless the NIC is cloned)/providing an extra layer to authentication process;  
data sent to a specific MAC address can only be accessed on that device; [3]
9. Data;  
A pointer/reference to the previous node;  
A pointer/reference to the next node; [3]
10. *Award up to [2 max].*  
A recursive call involves the use of stacks;  
For storing/pushing on/popping out data/ return addresses/return values *etc*;  
If many recursive calls are made, the memory usage can be very large; [2]
11. (Have sensors that) can react to external stimuli/ changes in its environment;  
without needing to contact any central/outside source/by making its own decisions/acts independently; [2]

## Section B

12. (a) (i) Data migration is the transfer of data from one system/storage device to another;  
(Importing the data/database from the company to the company system) [1]

- (ii) *Award [1] for identifying a problem that may arise and [1] for an expansion / elaboration up to [2 max].*

*Mark as 2 and 2*

Data loss;  
Due to transmission faults/lack of adequate storage;

Incompatible file formats;  
Which could lead to incomplete or incorrect data transfer;

Different file structures;  
Which will result in a mismatch of data, for example in customer records;

Validation rules differ between companies;  
Which could lead to inconsistent/incorrect results;

Different character sets might be used;  
Which could lead to inconsistent/incorrect results;

Different languages might be used;  
Leading to translation issues;

Data corrupted when transferring (data) files;  
And not usable at destination;

**Note:** *Accept data loss as an issue unless the expansion is a repeat of the second issue identified.*

[4]



- (b) *Award **[1]** for identifying each aspect of change management that may arise and **[1]** for an expansion / elaboration up to **[2 max]**.*

*Mark as 2 and 2*

Workforce issues;  
Such as redundancy/retraining;

The time frame involved;  
In merging the two systems;

Testing;  
Of the combined systems/new data;

Data entry;  
If migration not possible;

Costs involved;  
In the aligning of the two systems;

Changeover decisions;  
Such as parallel running;

**[4]**

- (c) (i) Award up to **[3 max]**.

**If intermediate columns are not used:**

If A, B and C are all correct, award 3 marks

If A, B and three outputs from C are correct, award 2 marks

If A and B are correct, award 1 mark

**If intermediate columns are used:**

If C is not correct and intermediate columns are used:

Award **[1]** mark for Columns A and B being correct

Award **[1]** mark for either X or Y or both being correct

If A, B and C are all correct, award 3 marks

A	B	NOT(A AND B)	A OR B	C=NOT(A AND B) AND (A OR B)
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	0	1	0

**[3]**

- (ii) XOR gate;

**Note:** Follow through from an incorrect truth table in part (c)(i).

Accept  $A \text{ XOR } B = C$

**[1]**

- (d) Two truth tables can be compared;

The same outputs from the same inputs, the circuits are equivalent;

**[2]**

13. (a) Award **[1]** mark if evident from the trace table that the loop executes exactly 4 times  
**[1]** mark for correct final contents of B (2 and 4)  
**[1]** mark for the correct final contents of C (3)

DATA.hasNext ( ) ?	A	A >= 0 ?	A mod 2 = 0?	Contents of B	Contents of C
true	2	true	true	2	
true	4	true	true	2, 4	
true	-1	false	*true	*2, 4	
true	3	true	false	*2, 4	3
false					

**Note:** These cells (\*) could be left empty

**[3]**

- (b) Award **[4]** marks as follows:

A position/index of the middle value calculated from HI and LOW;  
 If search value equals to value in the array at this position, then end;  
 Otherwise, change HI or LOW according to whether search value is above or below;  
 Repeat the process until search value is found;

Example answer 1:

1. Calculate MID (the midpoint of LOW and HI)
2. If ARRAY[MID] is the search value, end, else
3. If search value < than ARRAY[MID], HI = MID-1, else LOW = MID+1
4. Repeat steps 1, 2 and 3 until found

Example answer 2:

Find/calculate the midpoint/median value between HI and LOW;  
 If the array value at midpoint equals the search value; then the search value is found;  
 If search value is greater than the array value at midpoint, set the midpoint as a new LOW and repeat the process - if search value is less than the array value at midpoint, set midpoint as a new HI;  
 Repeat until search value is found;

**[4]**

(c) Award up to **[5 max]** as follows:

Example answer 1:

**[1]** mark for a loop through collection NUMBERS

**[1]** mark for the assignment into the array

**[2]** marks for a correct linear search, **[1]** mark for the loop and 1 mark for the comparison and use of flag

**[1]** mark for correct input of search value and output message

```
input S // search item
COUNT = 0
loop while NUMBERS.hasNext()           // or ... while NOT
                                        //NUMBERS.isEmpty()
    D[COUNT] = NUMBERS.getNext()
    COUNT = COUNT + 1
end loop
FOUND = false
X = 0
loop while (X < COUNT) and (FOUND == false) //accept for loop
    if S == D[X]
        FOUND = true
        X = X + 1
    end if
end loop
if FOUND == true
    output "found"
else
    output "not found"
end if
```

Example answer 2:

(in this example assignment into array done in same loop as reading from NUMBERS)

**1 mark** for a loop through collection NUMBERS

**1 mark** for the assignment into the array(**do not award this mark if flag is used to terminate/break the while loop**)

**2 marks** for correct linear search (through D), 1 mark for correct comparison, 1 mark for flag

**1 mark** for correct input of S and output message

```
input S // search item

COUNT = 0
FOUND = false
loop while NUMBERS.hasNext()           // or ... while not NUMBERS.isEmpty()
    D[COUNT] = NUMBERS.getNext()
    if S == D[COUNT]
        FOUND = true
    end if
    COUNT = COUNT + 1
end loop
if FOUND
    output "found"
else
    output "not found"
end if
```

- (d) Award up to **[3 max]**.  
*[1] mark for setting and changing a flag*  
*[1] mark for starting from the 2nd entry and checking all values as being entered*  
*[1] mark for comparing the current value with the previous one*

**Note:** Do not accept pseudocode/code.

**[3]**

14. (a) *Award up to [2 max].*

Paging allows similar-sized sections of data;  
identified by specific addresses;  
To be swapped between RAM and secondary memory/storage;

Example answer:

Main memory is divided into page files/swap files/sections so that data from these sections can be transferred to and from secondary storage. The addresses of these sections in main memory are linked to virtual addresses in secondary memory and the links stored in a table. When a page is needed, the table is accessed to recall the page into the correct locations in main memory. [2]

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

The listed software would require a lot more RAM to run efficiently;  
Paging/VM would be regularly used;  
In order to load the required modules;  
Thus slowing down the operation of the laptop;  
The laptop does not have a dedicated graphics processor;  
Which would be required for a multimedia game;  
This would result in low quality graphics; [3]

(c) A stack is a LIFO data structure;

*In the game it allows the “undo” function to work / movements/steps can be retraced;*

**Note:** Allow any reasonable answer for the second marking point. [2]

- (d) Award up to **[6 max]** as follows.

Example answer 1:

Award **[1]** for while loop;

Award **[2]** for the correct condition in while statement (stack is not empty **and** queue is not full) award **[1]** in case of the minor error;

Award **[1]** for pop. (**Note:** Do not allow alternatives)

Award **[1]** for enqueue; (**Note:** Allow alternatives)

Award **[1]** for the output only if stack is not empty after the loop;

```
loop while NOT S.isEmpty() and NOT Q.isFull()
  N = S.pop()
  Q.enqueue(N)
end loop
if NOT S.isEmpty()      //accept NOT S.isEmpty() and Q.isFull()
  output "incomplete process"
end if
```

Example answer 2:

Award **[1]** for the correct use of flag throughout.

Award **[1]** for the while loop

Award **[1]** for checking `!(S.isEmpty())`

Award **[1]** for pop. (**Note:** Do not allow alternatives.)

Award **[1]** for checking if Q is full

Award **[1]** for enqueue; (**Note:** Allow alternatives.)

Award **[1]** for output

```
FLAG = true
loop while !(S.isEmpty())
  N = S.pop()
  if !(Q.isFull())
    Q.enqueue(N)
  else
    FLAG = false
    break
    // can be missed out or alternative used
  end if
end loop
if (FLAG = false)
  output "incomplete process"
end if
```

**[6]**

**Note:** Be lenient with the positioning of the “!” and use of parentheses and dot notation.

- (e) Award **[2]** marks, **[1]** for advantage, **[1]** for expansion.

A dynamic queue would not have predetermined/fixed size;

Memory is allocated as required/efficient use of memory/flexible size//there is always sufficient memory to accommodate the queue;

**[2]**

15. (a) 168; [1]  
**Note:** Award [1] for “24 x 7” seen.

- (b) Award up to [4 max] as follows:

Example answer 1:

Award [1] for any nested loop.

Award [1] for the correct nested loop.

Award [1] for the correct assignment to A.

Award [1] for the correct retrieval from WEATHER.

```
loop for DAY from 0 to 6
  loop for HOUR from 0 to 23
    A[DAY][HOUR] = WEATHER.getNext()
    // A[DAY][HOUR] = WEATHER.getData()
  end loop
end loop
```

Example answer 2:

Award [1] for initialization of POS and correct increment within the loop

Award [1] for the correct loop.

Award [1] for the correct calculation of DAY

Award [1] for the correct calculation of HOUR

Award [1] for the correct assignment to A( correct retrieval from WEATHER )

```
POS=0
loop while WEATHER.hasNext() //accept not WEATHER.isEmpty()
  DAY=POS div 24
  HOUR=POS mod 24
  POS=POS+1
  A[DAY][HOUR]=WEATHER.getData() // WEATHER.getNext()
end loop
```

[4]



(c) Award up to **[6 max]** as follows:

Example answer 1:

Award **[1]** for initialization of HIGHEST

Award **[1]** for initialization of NAMES

Award **[1]** for the correct outer loop

Award **[1]** for the correct inner loop

Award **[1]** for the correct comparison and the assignment of HIGHEST

Award **[1]** for the assignment of MAX\_DAY within if statement

Award **[1]** for the correct output statement

```
HIGHEST = A[0][0]           //accept for example HIGHEST = -100
NAMES=["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
// array NAMES may contain other names, like "Monday", "Tuesday", etc.
loop for DAY from 0 to 6
  loop for HOUR from 0 to 23
    if A[DAY][HOUR] > HIGHEST
      HIGHEST = A[DAY][HOUR]
      MAX_DAY = DAY
    end if
  end loop
end loop
output(NAMES[MAX_DAY])
```

Example answer 2:

Award **[1]** for initialization of HIGHEST.

Award **[1]** for the correct outer loop

Award **[1]** for the correct inner loop

Award **[1]** for the correct comparison and assignment of HIGHEST

Award **[1]** for the assignment MAX\_DAY within if statement

Award **[1]** for the if statement after the outer loop (accept switch statement)

Award **[1]** for the correct output statement (may be written within if statement)

```
HIGHEST = A[0][0]   //accept for example HIGHEST=-1000
loop for DAY from 0 to 6
  loop for HOUR from 0 to 23
    if A[DAY][HOUR] > HIGHEST
      HIGHEST = A[DAY][HOUR]
      MAX_DAY = DAY
    end if
  end loop
end loop
if MAX_DAY == 0 then
  D = "Monday"
else if MAX_DAY == 1 then
  etc...
end if
output D
```

**[6]**

(d)

*Award up to [4 max].*

**Note:** *Do not give any marks if the proposed structure is not dynamic.*

A binary search tree could hold a series of nodes each of which contains the data for each reading;

and two pointers (to the left and right child);

Each node is inserted according to the temperature, higher temperature to the left, lower to the right;

An in-order traversal would output the data sorted in descending order of temperature;

**Note:** Linked list would not minimise processing but should get credit.

*Award up to [2 max], [1] mark for correct explanation on how it can be constructed and [1] mark for correct explanation on how it can be used to output data in descending order;*

**[4]**

16. (a) *Award up to [2 max].*  
 An embedded system is a system that performs a specific/dedicated function;  
 For a larger system for which it is physically part of:  
 In this example, the control system is actually part of the electric motor; [2]
- (b) To control the output (values);  
 By keeping it close to the reference/input value;
- OR**
- To control the system performance/behaviour of the system/;  
 by routing back the output values as input values;  
 (which are used to make decisions on changing/modifying the performance of the system/behaviour of the system); [2]
- (c) *Award up to [6 max].*  
 The desired speed is set/inputted and the (speed) sensor / input transducer measures the (current output) speed;  
 This analog signal (from the speed sensor) is converted into digital;  
 And sent back to the controller/processor;  
 The error/difference between the output speed and desired/input speed is calculated and compared to pre-set values to determine whether speed should be adjusted;  
 If the speed/power should be adjusted by an amount dependent upon the error then signal is sent to the output (transducer);  
 The above process is repeated continually; [6]
- (d) The sensor sends an interrupt to the operating system;  
 When it measures a value outside of the normal range;  
 Such as an abnormal temperature/pressure;  
 The operating system sends a signal to operate an actuator (output transducer);  
 In order to bring the system back to normal; [5]
-