



# **MARKSCHEME**

**May 2008**

**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 IB Cardiff.*

## General Marking Instructions

*After marking a sufficient number of scripts to become familiar with the markscheme and candidates' responses to all or the majority of questions, Assistant Examiners (AEs) will be contacted by their Team Leader (TL). The purpose of this contact is to discuss the standard of marking, the interpretation of the markscheme and any difficulties with particular questions. It may be necessary to review your initial marking after contacting your TL. **DO NOT BEGIN THE FINAL MARKING OF YOUR SCRIPTS IN RED INK UNTIL YOU RECEIVE NOTIFICATION THAT THE MARKSCHEME IS FINALIZED.** You will be informed by e-mail, fax or post of modifications to the markscheme and should receive these about one week after the date of the examination. If you have not received them within 10 days you should contact your TL and IB Cardiff. Make an allowance for any difference in time zone before calling. **AEs WHO DO NOT COMPLY WITH THESE INSTRUCTIONS MAY NOT BE INVITED TO MARK IN FUTURE SESSIONS.***

You should contact the TL whose name appears on your “Allocation of Schools listing” sheet.

### **Note:**

Please use a personal courier service when sending sample materials to TLs unless postal services can be guaranteed. Record the costs on your examiner claim form.

## General Marking Instructions

1. Once markscheme is received mark in pencil until final markscheme is received.
2. Follow the markscheme provided, do **not** use decimals or fractions and mark only in **RED** once mark
3. Where a mark is awarded, a tick (✓) should be placed in the text at the **precise point** where it becomes clear that the candidate deserves the mark.
4. Sometimes, careful consideration is required to decide whether or not to award a mark. Indeed, another examiner may have arrived at the opposite decision. In these cases write a brief annotation in the **left hand margin** to explain your decision. You are encouraged to write comments where it helps clarity, especially for moderation and re-marking.
5. Unexplained symbols or personal codes/notations on their own are unacceptable.
6. Record subtotals (where applicable) in the right-hand margin against the part of the answer to which they refer. Show a mark for each part question (a), (b), *etc.* Do **not** circle sub-totals. Circle the total mark for the question in the right-hand margin opposite the last line of the answer.
7. Where an answer to a part question is worth no marks, put a zero in the right-hand margin.
8. **Section A:** Add together the total for the section and write it in the Examiner Column on the cover sheet.  
**Section B:** Record the mark awarded for each of the six questions answered in the Examiner Column on the cover sheet.  
**Total:** Add up the marks awarded and enter this in the box marked TOTAL in the Examiner Column on the cover sheet.
9. After entering the marks on the cover sheet check your addition of all marks to ensure that you have not made an arithmetical error. Check also that you have transferred the marks correctly to the cover sheet. **We have script checking and a note of all clerical errors may be given in feedback to all examiners.**
10. Every page and every question must have an indication that you have marked it. Do this by **writing your initials** on each page where you have made no other mark.
11. A candidate can be penalized if he/she clearly contradicts him/herself within an answer. Once again make a comment to this effect in the left hand margin.

## Subject Details: Computer Science HL Paper 1 Markscheme

### Mark Allocation

Section A: Candidates are required to answer **all** questions. Total 40 marks.

Section B: Candidates are required to answer **all** questions. Total 60 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.
- 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.
- The order of statements does not have to be as written (unless stated otherwise).
- 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 penalising 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**”.

**SECTION A**

**Total: [40 marks]**

1. *Award up to [3 marks max].*  
 Description of proposed system;  
 Estimated costs;  
 If it is economically viable;  
 If it is technically possible;  
 Any legal issues;  
 Estimated completion date / time for completion; **[3 marks]**
  
2. *Award up to [4 marks max].*  
 Original specifications may have changed;  
 Requiring an updated system;  
  
 Errors in software;  
 Requiring software to be rewritten;  
  
 New technology becomes available;  
 That would improve (the efficiency) of the system; **[4 marks]**
  
3. *Award up to [3 marks max].*  
 Valid/correct;  
 Invalid/incorrect;  
 Borderline/equivalent explanation;  
 Normal / Abnormal/Extreme; **[3 marks]**
  
4. *Accept **either** answer. Award up to [2 marks max].*  
 A compiler produces a complete object code program (which can be saved);  
 Whilst an interpreter does not;  
  
 An interpreter runs each block of code as soon as it is translated;  
 Whilst a compiler cannot run the object code until the whole program is translated; **[2 marks]**
  
5. Direct access allows for the location/retrieval/updating/processing of a data item/record;  
 Without having to access/look at other data items/records first; **[2 marks]**
  
6. *Award [1 mark] for an appropriate application and [1 mark] for a suitable expansion (e.g. a brief description of how it would be used).*  
*Example:*  
 Cash-point machine;  
 Where the customer works his/her way through the various menus by touching the appropriate parts of the screen in order to withdraw money; **[2 marks]**

7. (a) B6; [1 mark]
- (b) 10110110; [1 mark]
8. Underflow occurs when the result of a **calculation** is too small;  
To be stored (in the given number representation); [2 marks]
9. Award [2 marks] for correct answer. Award [1 mark] for  $A.B + B$ , or for an incorrect initial step followed by a simplification.  
B [2 marks]
10. (a) Award up to [2 marks].  
Contains the instruction;  
That is (currently) being executed; [2 marks]
- (b) Award up to [2 marks max].  
Working area (of the processor);  
(Temporary) store for data;  
Resulting from (intermediate stages of ) calculations; [2 marks]
11. (a) Award up to [2 marks max].  
The algorithm calls itself;  
There is a “simple”/terminating case;  
There is a “more difficult” case that becomes increasingly “simpler”; [2 marks]
- (b) Award [2 marks] for correct answer.  
8 [2 marks]
12. Award up to [2 marks max].  
Input data is required immediately by the process/program;  
Several inputs/events are happening simultaneously;  
Responses are time-critical / system must be fail-safe; [2 marks]
13. (a) Award [1 mark] for correct answer.  
4E [1 mark]
- (b) Award [2 marks] for correct answer.  
11101100 [2 marks]

14. Retrieval time for data is reduced;  
Because files are (re-arranged so that they are) contiguous; **[2 marks]**
15. *Award up to [2 marks max].*  
Feature in which the same operation;  
Can be applied to different objects/objects of different types;  
Correct use of terms overloading and overriding; **[2 marks]**
16.  $O(n \log n)$  **[1 mark]**



**SECTION B**

**Total: [60 marks]**

17. (a) *Award up to [3 marks max].*  
 If records are sorted they can be added/deleted more easily;  
 Without having to re-sort/shuffle the records;  
 Just by manipulating the pointers;  
 Better use of memory;  
 As records can be (dynamically) added/deleted;  
 No wasted space (as in an array); **[3 marks]**
- (b) *Award up to [3 marks max].*  
 Guillen node is created;  
 Temporary pointer points to the Guillen node;  
 The place to insert is found (by traversing the list);  
 Guillen node points to the node that the node before the insertion point points to;  
 Node before insertion point now points to Guillen node; **[3 marks]**
- (c) (i) (Binary) tree; **[1 mark]**  
*Do not accept arrays, doubly linked list.*
- (ii) *Award up to [3 marks max]. Award up to [2 marks max] if only advantages or disadvantages are given.*  
 Faster retrieval of data;  
 Especially for large number of nodes/data items;  
 As less nodes to search;  
 Particularly if tree is well-balanced;  
 Some operations are more complicated to program than for a linked list;  
 e.g. deletion; **[3 marks]**

**Total: [10 marks]**

18. (a) Faster to access individual record than by other types of access;  
No need for sequential access as only individual records are looked up (normally); **[2 marks]**
- (b) *Award up to [2 marks max].*  
The hashing algorithm would use the key field;  
Which will form part of the calculation;  
That will determine the address where each record will be stored; **[2 marks]**
- (c) *Award up to [2 marks max].*  
So that it is fast to calculate;  
To minimise the number of collisions;  
To match the number of possible values to the memory space available;  
To provide enough locations as there are records; **[2 marks]**
- (d) (i) 231; **[1 mark]**
- (ii) *Award up to [3 marks max].*  
Would generate the same address;  
Which would create a collision;  
The record would be stored in the next available free space;  
Which would be pointed to from the original address;  
Overflow and chaining; **[3 marks]**

**Total: [10 marks]**

19. (a) The return address would be stored in the stack;  
This address would be popped back into the program counter;  
Once the subprogram has been completed; **[3 marks]**
- (b) To allow the use of stacks;  
When evaluating expressions; **[2 marks]**
- (c) *Award [2 marks] for correct answer.*  
 $AB + C$  / **[2 marks]**
- (d) *Award [3 marks] if diagram clearly explains the following sequence.*  
A is pushed onto the stack followed by B;  
When the operand is encountered the expression is evaluated / A is added to B;  
The result is pushed back onto the stack; **[3 marks]**
- Total: [10 marks]**

20. (a) *Award [1 mark] for a suitable example.*  
*e.g. processor sending data to a printer;*  
*Award [1 mark] for the concept of a “conversation” taking place.* **[2 marks]**
- (b) Modem / router / gateway; **[1 mark]**
- (c) The firewall will prevent the access;  
From certain sites;  
Depending upon the parameters previously established; **[3 marks]**
- (d) *Award [1 mark] for the security procedure and [1 mark] for a reason for employing this measure (not just a description), for up to **two** different procedures.*
- Data should be encrypted;  
Allowing only those with decrypting code access;
- (Safe) practices should be part of staff training;  
Which will create a security conscious culture;
- User names/passwords employed / should be regularly changed/made difficult to guess;  
So that no access is possible without them;
- Installation of virus checkers/spyware software;  
To prevent damage to files or the system / to prevent data being extracted from the files/system;
- [4 marks]**
- Total: [10 marks]**

21. (a) XOR; [1 mark]

- (b) Award [1 mark] for the correct `if` statement.  
Award [1 mark] for the correctly placed returns.

```
public boolean logic2(int a , int b)
{ if ((a==1)&&(b==1)) return false
  else return true;
}
```

[2 marks]

- (c) Award marks as follows.  
[1 mark] for correct return type in the signature.  
[1 mark] for use of nested loops.  
[1 mark] for correct loop conditions.  
[1 mark] for correct comparison.  
[1 mark] for values returned correctly.

```
public boolean compareCircuits()
{
    boolean check = true;
    for(int a = 0; a < 2; a++)
    { for(int b= 0; b < 2; b++)
      { if(!(logic1(a,b) == logic2(a,b))
        { check = false;
          return check;
        }
      }
    }
    if (check = true)
        return check;
}
```

[5 marks]

- (d) Award [2 marks] for the final answer.  
Award [1 mark] if final answer is incorrect, but some correct simplification has been shown.

$$\begin{aligned}
 & \overline{A} \cdot \overline{B} + \overline{A} \cdot B + A \cdot \overline{B} \\
 &= \overline{A} \cdot (\overline{B} + B) + A \cdot \overline{B} \\
 &= \overline{A} \cdot (1) + A \cdot \overline{B} \\
 &= \overline{A} + A \cdot \overline{B} \\
 &= \overline{A} + \overline{B} \\
 &= \overline{A \cdot B}
 \end{aligned}$$

[2 marks]

**Total: [10 marks]**

22. (a) *Award up to [2 marks max].*  
Allocation/deallocation of space (on the backing store) for each file;  
Store/maintain access rights;  
Hold the address (of the different parts) of each file;  
Maintain archive information of each file (e.g. author, dates of creation/last update);  
*[2 marks]*
- (b) *Award [1 mark] each for the correct labelling of the track, sector and cylinder.* *[3 marks]*
- (c) The heads are in the same position / reading the same track (on different disks);  
So if data is written to tracks within the same cylinder / (diagram that shows this);  
More data can be processed without moving the heads;  
*[3 marks]*
- (d) Blocking refers to the reading/writing of several items of data/records at the  
same time/as a group;  
Thus speeding up the reading/writing process;  
*[2 marks]*
- Total: [10 marks]*
-