



MARKSCHEME

November 2009

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**.
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 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 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. (a) Scanners convert/digitize hardcopy text or graphics;
Into computer usable form; **[2 marks]**
- (b) *Award up to [1 mark max].*
Possible answers:
Barcode reader (for example in supermarkets);
Optical mark reader (multiple choice tests);
Handwritten character reader (cheques);
etc. **[1 mark]**
2. *Award up to [3 marks max].*
Quiet;
Output of text and graphics;
Excellent quality;
(Very) high speed;
etc. **[3 marks]**
3. *Award up to [3 marks max].*
Network is collection of data communication hardware;
Computers;
Communications software;
Communications media;
Connected in a way to allow users to share resources; **[3 marks]**
4. *Award up to [2 marks max] × 3 = [6 marks].*
- Seek time** is the time it takes to find the proper track;
And place a read/write head over it;
- Rotational delay** is the time it takes to rotate the disk plate;
And position correct sector over the heads;
- Data transfer time** is the time it takes to transfer data;
From disk to RAM; **[6 marks]**

5. (a) *Award up to [2 marks max].*
An operating system is software;
That controls a computer;
Examples:
Provides an interface for users/programs/hardware;
Controls the organization and execution of all programs;
Manages memory;
etc. **[2 marks]**
- (b) *Award up to [3 marks max].*
Examples:
Easy to use;
Speed of operation;
Capability to support multitasking;
Compatibility (capability of being used with hardware);
Quantity of compatible software user needs to install;
Availability of technical support;
etc. **[3 marks]**
6. (a) binary tree / hash table; **[1 mark]**
- (b) queue; **[1 mark]**
- (c) stack; **[1 mark]**
7. $26/16 = 1$ remainder $10 = A$
 $1/16 = 0$ remainder 1
 $26_{(10)} = 1A_{(16)}$ **[1 mark]**
8. $11010_{(2)}$; **[1 mark]**
9. (a) GEABFCD **[2 marks]**
- (b) ABECDFG **[2 marks]**
- (c) AEBGCFD **[2 marks]**

Note: If any four elements in correct place award **[1 mark]**.

10. *Award up to [3 marks max].*

Ordering: parent-child relationships (direction of inheritance is one-way);

Use: the ability of subclasses to use fields and methods from the superclass which they have not defined themselves;

Overriding: ability of subclasses to redefine fields and methods in the superclass;

Possible answers:

A way of organizing classes and subclasses;

(Such that a superclass) can define data members and methods only once;

And yet apply to the classes which inherit (from the superclass);

[3 marks]

11. (a) A;

[1 mark]

(b) C;

[1 mark]

12. *Award up to [2 marks max].*

A technique for allowing one device to check the status of another;

At a regular interval;

To determine whether the second device is ready to send/receive data;

[2 marks]

13. *Award up to [2 marks max].*

Each bit of the interrupt register represents a different type of interrupt;

If the interrupt occurs, the corresponding bit is set;

At the beginning of machine instruction cycle the interrupt register is checked and, if a bit is set, the OS can call the appropriate interrupt handler;

[2 marks]

SECTION B

Total: [60 marks]

14. (a) *Award [2 marks] for each phase up to [8 marks max].*

Input

Data is put into computer;

And converted into a form that can be processed by the computer;

Processing

All the operations (calculations, logical operations) are done;

That are necessary to obtain the output results;

Storage

Computer stores data and programs;

In a computer usable form (on disk or any other storage media);

Output

The result of processing is provided to the user;

In the form usable by the user (hardcopy, displayed on the monitor, *etc.*);

[8 marks]

- (b) *Award [1 mark] for each comparison up to [2 marks max].*

Primary storage has

Faster access time;

Smaller capacity;

Greater cost;

Than secondary storage.

[2 marks]

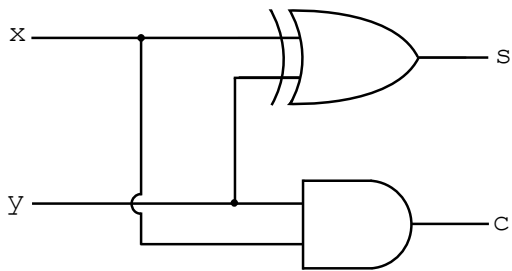
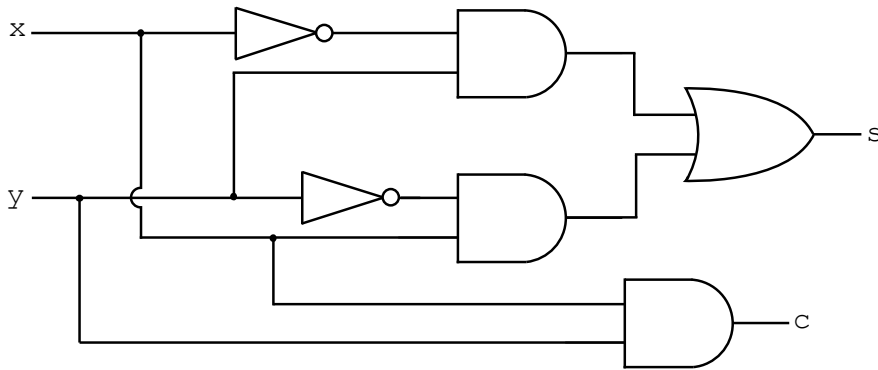
Total: [10 marks]

15. (a) (i) $c = x \cdot y$ [1 mark]

(ii) $s = x \cdot \bar{y} + \bar{x} \cdot y$ [2 marks]

(b) Award [1 mark] for showing two outputs and [1 mark] for each correct gate ([2 marks] if they have used an XOR gate for s) up to [4 marks max].

Examples:



[4 marks]

(c) Award [3 marks] for 7 or 8 correct rows, [2 marks] for 5 or 6 correct rows and [1 mark] for 4 correct rows, up to [3 marks max].

A	B	C	$\overline{A+B \cdot C}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

[3 marks]

Total: [10 marks]

16. (a) (i) Add an item;
On the top of the stack; *[2 marks]*
- (ii) Remove an item;
From the top of the stack; *[2 marks]*
- (b) *Stack overflow is a run-time error that occurs when*
Attempting to push an item;
Onto a full stack;
- Whilst stack underflow occurs when*
Attempting to pop an item;
From an empty stack; *[4 marks]*
- (c) *Award [1 mark] for any reasonable operation up to [2 marks max].*
Initialize stack;
Check whether the stack is full;
Check whether the stack is empty;
Count the number of items on the stack;
Return the top item without popping it;
etc. *[2 marks]*

Total: [10 marks]

17. (a) Award [1 mark] for appropriate variable name and data type up to [3 marks max].
Example:

```
public class Room
{
    String description;
    boolean occupied;
    double cost;
}
```

Award [1 mark] for each sample of appropriate data type up to [3 marks max].
Example:

```
Room x = new Room();
x.description = "two beds, air conditioner, internet connection";
x.occupied = true;
x.cost = 256.50;
```

Code is not requested, award marks for any correctly labeled diagram.

[6 marks]

- (b) (i) Award up to [2 marks max].

Examples:

One dimensional array;

Of objects/rooms/etc.;

Accept array declaration for [2 marks].

```
Room[] hotel = new Room[30];
```

Candidates can describe a dynamic data structure, such as

A linked list / binary tree;

Where each node represents a room;

[2 marks]

- (ii) Award up to [2 marks max].

All array elements have the same name, each one can be accessed directly;

By writing index/subscript;

If candidate's answer to (i) is linked list/binary tree, then accept a description of access to the corresponding data structure and award up to [2 marks max].

Or candidate may show, by example code, that data about k^{th} room can be accessed as follows:

```
hotel[k].description = "single bed, air conditioner";
hotel[k].occupied = true;
hotel[k].cost = 158.00;
```

[2 marks]

Total: [10 marks]

18. (a) (i) *Award up to [2 marks max].*
Data record is retrieved/accessed in order;
In which it was stored;
All preceding records should be read; *[2 marks]*
- (ii) *Award up to [2 marks max].*
Data record can be accessed directly;
Without reading all the preceding records;
By calculating its relative position within the file; *[2 marks]*
- (iii) Sequential access to index/table that holds key fields and
storage locations/addresses;
Direct access to data record by using address/storage location; *[2 marks]*
- (b) (i) *Award up to [2 marks max].*
To store key fields and addresses that are calculated from the primary keys;
For fast access/retrieval/storage of data records;
In an unordered file; *[2 marks]*
- (ii) Two or more primary keys (which are turned into address);
Are given the same address; *[2 marks]*

Total: [10 marks]

19. (a) *Award up to [3 marks max].*
 Actual parameter value is 7 ($n = 7$) and method returns a single value;
for loop executes two times;
 n is never divisible by the current value of j;
 Value returned is true;

Example answer:

n	j	temp	$j \leq n/2$	$n \% j == 0$
7				
	2	true	true	false
	3	true	true	false
	4		false	

[3 marks]

- (b) $O(n)$;

[1 mark]

- (c) *Award up to [2 marks max].*
 Algorithm returns true if n is a prime number;
 false otherwise;

or

Algorithm returns false if n is divisible by any value;
 In the range from 2 to $n/2$;
 true otherwise;

[2 marks]

- (d) (i) *Award up to [2 marks max].*
 A variable which is declared within method (or block/compound statement);
 And can be used only in that method (block);

Example:

temp is local to method check(), j is local to the **for** loop;

[2 marks]

- (ii) *Award up to [2 marks max].*
 Parameter used in method definition;
 It is given a value when method is called;

Example:

int n in method check();

[2 marks]

Total: [10 marks]