International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

MARKSCHEME

May 2012

COMPUTER SCIENCE

Higher Level

Paper 1

15 pages

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 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**".

SECTION A *Total: [40 marks]*

1. Early detection of errors in source code through syntax analysis;
Reusability/portability of programs across different platforms;
Conversion of source code to executable/object code; ***[2 marks]***

2. *Award up to **[2 marks max]**.*
CPU has more primary memory and more registers than a microprocessor;
CPU stores more programs in ROM than a microprocessor does;
CPU executes more programs (faster) than a microprocessor;
Input/output in some microprocessors can be analogue; ***[2 marks]***

3. *Award up to **[4 marks max]**.*
Validation aims to reduce data errors, especially at input stage;
Data not in a prescribed format for processing are discarded;
To prevent propagation of invalid data in the system;
Verification aims to establish the accuracy of the input data;
And limits errors introduced by human operators in inputting data; ***[4 marks]***

4. To manage the (fair) access to shared resources (processor time, memory, *etc.*);
Among several competing processes that run simultaneously;
To monitor the functioning of hardware resources and report malfunctions; ***[2 marks]***

5. (a) *Award **[1 mark]** for the correct integer part;*
*Award **[1 mark]** for the correct decimal part;*
01100010 ***[2 marks]***

(b) 10011110; ***[1 mark]***

6. *Award **[1 mark]** for clearing or erasing pre-existing data from the dis. Award up to **[2 marks max]**.*
An empty file system is set up (or multiple file systems are, if the disk is partitioned);
A boot sector is installed, with code for booting programs;
(That are possibly stored elsewhere in the disk;)
Accept more detailed descriptions on the role of blocks, block tables, booting sectors, etc. ***[2 marks]***

7. *Award up to **[4 marks max]**.*
The effect of a logic error is visible at run time;
And fixing it may require a global study of the program;
The **behaviour** of the program's code is incorrect;
Because the **sequence** of instructions and/or **choice** of condition is incorrect;

The effect of a syntax error is visible at compile time;
And these errors can be automatically detected and locally fixed;
As they are found by **compiler/interpreter** during syntax analysis; ***[4 marks]***

8. (a) $2^5 \times 2^5 \times 2^5$ $(= 32768)$; ***[1 mark]***

(b) $70 \times 100 \times 2$ Bytes $= 14\,000$ Bytes (or 14 kB, or $14\,000 \div 1024 = 13.67$ kiB); ***[1 mark]***

(c) *Award **[1 mark]** for each modality, up to **[2 marks max]**.*
Choose a different colour representation;
And augment the number of pixels to be stored in 2 bytes;

Resize the initial dimensions;
To store with the same colour encoding a smaller number of pixels;

Accept bitmap and explanation of how it works.
Do not accept compression only, without explanations. ***[2 marks]***

9. *Award **[1 mark]** per feature up to **[2 marks max]**.*
Power settings which allow a computer to be put to sleep;
Cheaper LCD screens which consume less energy;
And offer better visual resolution, so reducing need to print;
Hardware parts of modern computers are made of environmentally friendly materials(less acidic, recyclable)
Fast network and data transfer rate to support teleworking/teleconferencing; ***[2 marks]***

10. (a) *Award **[1 mark]** for each of the correct subformulas of the main +, up to **[2 marks max]**.*

O = $\overline{A}.B + \overline{\overline{A} + \overline{A} + B}$

Accept other notations.

$O = (\neg A \wedge B) \vee \neg(\neg A \vee (\neg A \vee B))$ ***[2 marks]***

(b) *Award marks as follows up to **[2 marks max]**.*
*Award **[1 mark]** for the intermediate simplification;*
*Award **[1 mark]** for the simplified formula;*

$\overline{A}.B + A.\overline{\overline{A} + B} =$
$= \overline{A}.B + A.A.\overline{B} =$
$= \overline{A}, B + A.\overline{B}$

OR

$\overline{A}.B + .\overline{\overline{A} + B}$
$= \overline{A}.B + A.\overline{B}$

Accept other notation.
For example:

$(\neg A \wedge B) \vee \neg(\neg A \vee (\neg A \vee B)) =$
$(\neg A \wedge B) \vee \neg(\neg A \vee B) =$ (associativity + idempotency)
$(\neg A \wedge B) \vee (A \wedge \neg B) =$ (de Morgan law)

$(\neg A \wedge B) \vee \neg(\neg A \vee (\neg A \vee B)) =$
$(\neg A \wedge B) \vee (A \wedge (A \wedge \neg B)) =$ (de Morgan law)
$(\neg A \wedge B) \vee (A \wedge \neg B) =$ (associativity + idempotency) ***[2 marks]***

(c) *Award up to **[2 marks max]**.*

Example answer 1:
Build the truth table for A **xor** B / define A **xor** B;
Compare it line by line with the one for O in part (a) / the expression obtained in part (b);

Example answer 2:
The expression obtained in part (b) is logically equivalent to O;
And is itself the definition of the **xor** logical relation; ***[2 marks]***

11. *Award **[1 mark]** for each advantage up to **[2 marks max]**.*
Simpler/cheaper to implement (no cost of synchronisation as in par. trans.);
More adequate for long-haul transmissions;
Transfer rate faster (less signal processing is needed);
Signal integrity better (cheaper to implement); ***[2 marks]***

12. *Award marks as follows up to **[4 marks max]**.*
*Award **[1 mark]** for the disadvantage;*
*Award **[1 mark]** for the explanation;*
*For **two** cases.*

Time consuming;
Because several persons with several roles might be interviewed;
And interviews must target these different categories of people;

Interviewees may talk about different things, making comparisons difficult;

Answers may not reflect the expectation of persons in roles different from the one of the interviewee;
This can result in contradictory or inconsistent data;

People are often biased/untruthful in interviews;
This can produce incorrect data that can mislead the design team;

Possible not homogeneous interpretations of collected data;
Interviewers might value some details more than others; ***[4 marks]***

13. *Award **[1 mark]** for each of the **three** pieces of information, up to **[3 marks max]**.*

Sender's IP address;
Receiver's IP address;
Packet number;
Total number of packets;
Data;
Parity bit / check digit; ***[3 marks]***

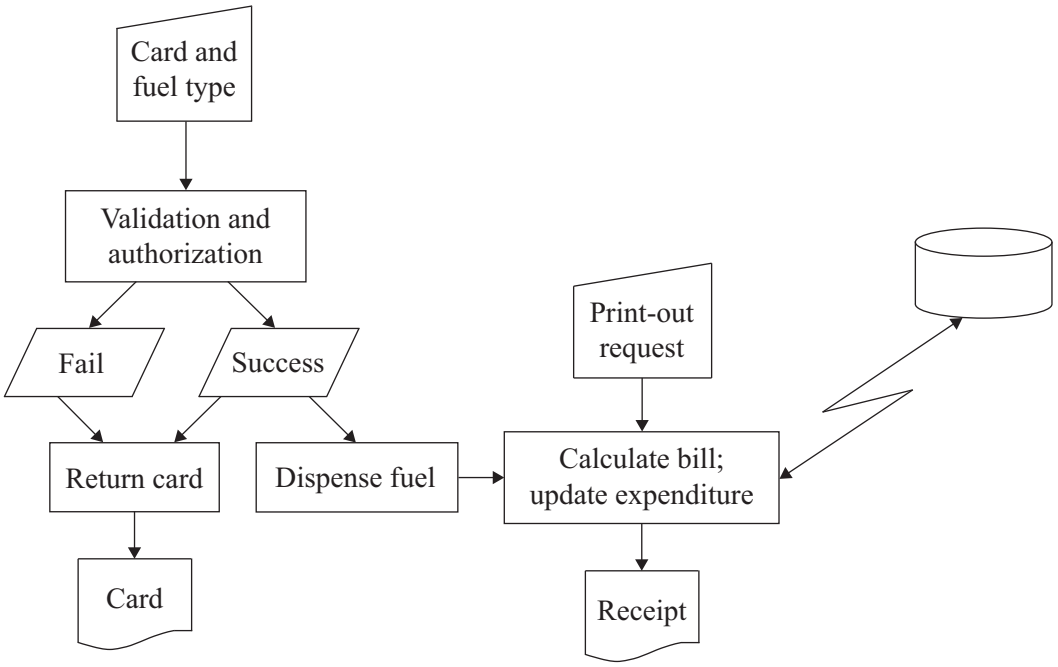
SECTION B

Total: [60 marks]

14. (a) Card's details (company/number) extracted from chip;
Send request of authorization to company transmitting card number, pin, public key and (limit of) 70 EUR and waits for reply message; ***[2 marks]***

(b) The current expenditure; ***[1 mark]***

(c) *Award marks as follows up to **[4 marks max]**.*
*Award **[1 mark]** for the two inputs (Card and fuel type)*
*Award **[1 mark]** for process of calculating bill and producing output;*
*Award **[1 mark]** for indicating access/communication to store for update;*
*Award **[1 mark]** for returning the card in both failed/success authorization;*
*Award **[1 mark]** for dispensing fuel;*
*Award **[1 mark]** for card validation/transaction;*



Different styles for drawing flowcharts are acceptable. Award marks whenever the criteria stated above are clearly visible in the drawing. . ***[4 marks]***

(d) *Award up to **[3 marks max]**.*
If the customer is attacked;
A robber can fill his car instead, and also steal the card;
But the limit of 70 EUR minimizes the loss for the customer; ***[3 marks]***

Total: [10 marks]

15. (a) (i) `n`; ***[1 mark]***

(ii) `y`; ***[1 mark]***

(b) *Award marks as follows up to **[2 marks max]**.*
*Award **[1 mark]** for correct **`n`** and **`y`** (both) columns;*
*Award **[1 mark]** for correct **`return`** column;*
*Award **[1 mark]** for correct last line (**`y`** empty, **`int`** value in **`return`** column);*

n	**y**	**return**
3	1	foo(10)
10	0	foo(5)
5	1	foo(16)
16	0	foo(8)
8	0	foo(4)
4	0	foo(2)
2	0	foo(1)
1		1

[2 marks]

(c) *Award up to **[2 marks max]**.*
The fragment of code for `n > 1` appears first (*e.g.* within a `while` loop or `if` statement);
And the code for `n = 1` and `n <= 0` follows (as exit from the loop/conditional);

Possible code answer 1:

```
public int foo(int n)
{
  while (n > 0)
  {
    int y = n % 2;
    if (y == 0)    return foo(n / 2);
    else           return foo(3 * n + 1);
  }
  if (n == 1) return 1
  else return -1;  // or symmetrical test
}
```

Possible code answer 2:

```
public int foo(int n)
{
  if (n > 0)
  {
    int y = n % 2;
    if (y == 0)    return foo(n / 2);
    else           return foo(3 * n + 1);
  }
  if (n == 1) return 1
  else return -1;  //  or symmetrical test
}
```

[2 marks]

continued …

Question 15 continued

(d) *Changes to* `foo` *are:*
New method signature: `public boolean bfoo(int n);`
Return values `false`/`true` instead of `-1`/`1` **and** recursive call on `bfoo`; ***[2 marks]***

(e) *Award* ***[1 mark]*** *for an advantage and* ***[1 mark]*** *for a disadvantage, up to* ***[2 marks max]****.*

Possible advantages:
More elegant way to express the solution, for some problems;
Recursive algorithms reflect mathematical/structural definitions;
Recursive programs have shorter code;

Possible disadvantages
A recursive method needs more memory than its iterative counterpart;
Time performance may be worse (because it freezes recursive calls);
Debugging a recursive code can be tricky, so maintenance is difficult; ***[2 marks]***

Total: [10 marks]

16. (a) Because CPU would periodically check all sensors in the network;
CPU time is wasted if there is no need of action, and it could be assigned to other processes; ***[2 marks]***

(b) *Award marks along these guidelines up to* ***[3 marks max]****.*
Interrupt signal is sent to computer;
CPU activates the event handler and suspends/puts on stack current tasks;
CPU services the tasks related to the request of interrupt;
When these finish, CPU resumes suspended tasks/pops them from stack; ***[3 marks]***

(c) *Award up to* ***[1 mark max]****. Possible answers:*
Sending an interrupt signal to the computer;
Comparing the sensed values with the admitted values' range; ***[1 mark]***

continued …

Question 16 continued

(d) *Award marks as follows up to **[4 marks max]**.*
*Award **[1 mark]** for inferring information from the interrupt;*
*Award **[1 mark]** for activating/releasing actuators;*
*Award **[1 mark]** for a program that depends on current values;*
*Award **[1 mark]** for polling (or similar mechanism) for a feedback;*
*Award **[1 mark]** for both activate and deactivate actuators (resource handling);*

Loop:
It identifies (through the interrupt) where/which anomalous value occurs;
It consequently decides which actuators to activate **and** executes the program to progressively modify values depending on the current values;
It polls the same sensor(s) for a new reading and a new interrupt may be generated;

Exit Loop:
The polled sensors do not reply with an interrupt;
The server deactivates/releases/puts in stand-by the actuators; ***[4 marks]***

Total: [10 marks]

17. (a) The length for each data type, for each field, is fixed (for both primitive/ reference types); ***[1 mark]***

(b) Primitive type `int` has already fixed length;
Assign to each `String` field a fixed number of bytes;
By adding extra spaces if necessary; ***[3 marks]***

(c) Records not frequently accessed implies slow disk access is still acceptable;
Fixed length makes the retrieval of desired records easy;
Records do not need to be ordered on disk; ***[3 marks]***

(d) *Award up to **[3 marks max]**.*
Answers may have a varying level of detail.
Award marks that highlight the following structure:

Solution Indexed file
Create an index (*e.g.* full);
Each record is associated to a unique value in the index;
The index can be searched sequentially and the record retrieved directly;

Solution Hashing
Define a hashing function based on string keys (to minimise collisions);
That returns the address of the record to be used in a hash map/array;
And have a mechanism for handling collisions; ***[3 marks]***

Total: [10 marks]

18. (a) In-order traversal (left-root-right); ***[1 mark]***

(b) Silvia is the left child of the sub-tree rooted Wei; ***[1 mark]***

(c) *Award up to **[4 marks max]**.*
Because the tree is unbalanced / was created by inserting only one node at a time;
Searching for a node/insertion point may require traversing the whole tree;
e.g. searching for Wei / inserting a node that is right child of Wei / the worst case is a right descending tree;
Hence, BigO efficiency of searching is linear/O(n), where n is the number of nodes; ***[4 marks]***

(d) *Award up to **[4 marks max]**.*
Several answers, more or less formal, are possible. They may be simplifications of the following examples:

Example answer 1: simple deletion, maintains the tree unbalanced
Find the node Marion;
Look in its left sub-tree to find the maximum (Lou) / find the maximum of Marion's minorants;
This node replaces Marion;
If it was not a leaf (*i.e.* it is the root of a left sub-tree), its sub-tree will be lifted up one level;

Example answer 2: deletion aiming in balancing the tree
Find the node Marion;
Look in its right sub-tree to find the minimum (Neil) / find the minimum of Marion's majorants;
This node replaces Marion;
If it was not a leaf (*i.e.* it is the root of a right sub-tree), its sub-tree can be lifted up one level;

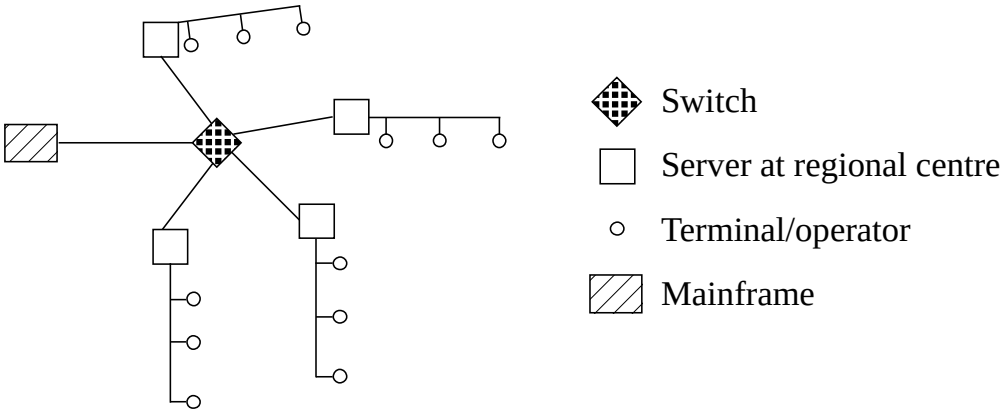
*Award **[2 marks]** for the correct drawing of the resulting tree alone.*

Do not give full marks if it is not evident that the result of deleting a node is indeed a structurally valid tree. ***[4 marks]***

Total: [10 marks]

19. (a) *Award marks as follows up to **[5 marks max]**.*
*Award **[1 mark]** for showing the mainframe;*
*Award **[1 mark]** for correct use of switch and possible gateways (to connect LAN to WAN);*
*Award **[1 mark]** for showing the servers at regional centres;*
*Award **[1 mark]** for correctly depicting the elements of the LANs;*
*Award **[1 mark]** for clearly labelling the elements;*

Several answers are possible: here it is depicted as a star-of-bus. Bus-of-stars (Bus as backbone), star-of-stars (with several switches) and ring-star are viable alternatives.



[5 marks]

(b) (i) Optical cable, general telecom lines; ***[1 mark]***

(ii) Twisted pair, coaxial cable;
Accept "air" and "Ethernet cable" but do NOT accept "Ethernet" without the "cable" modifier. ***[1 mark]***

(c) Because a switch inspects the data packet received (destination);
It forwards the packet to the appropriate server only;
Hence the performances of the network are better than with a hub that would broadcast; ***[3 marks]***

Total: [10 marks]