

# **MARKSCHEME**

**May 2005**

## **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 authorisation of IBCA.*

If you do not have a copy of the current Computer Science Guide,  
please request one from IBCA.

## 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 any 4 questions (10 marks each). Total 40 marks.  
Maximum total = 80 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 part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

- Each marking 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.
- The order of points 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 mark scheme 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. Effective communication is more important than grammatical niceties.
- 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

1. Award **[1 mark]** for each appropriate item and **[1 mark]** for its importance. Two items asked for.

For example:

System flowchart, which can be used later to ensure data flow is as required and adapted in future developments.

Algorithms developed for specific parts of system, which can be adapted later in case of error or need to update.

Annotated code, e.g. explaining purpose of each procedure, to identify any problems and to aid in maintenance.

*Do not accept feasibility studies, analysis of problem etc. unless they are clearly explained in relation to the software life cycle.*

For example:

Original/system design so that it can be examined to see if it meets the specifications and how it relates to the code for future evaluation, maintenance and modifications.

Allow program code and don't penalise for not writing annotated.

**[4 marks max]**

2. Award **[1 mark]** for suitable use and **[2 marks]** for explanation as to why suitable. Only award full marks for the explanation if it shows clear understanding of OMR

For example:

Completing multiple choice questionnaires / examinations. These can be prepared by computer, completed in pencil and used as input documents that are then analysed. Hence the human labour / cost is reduced.

Completing gas/electricity readings from meter. All details on client and previous readings printed out, completed by reader or client, input to computer and used to produce bill.

**[3 marks max]**

3. Award **[1 mark]** for base difference and **[1 mark]** for elaboration.

Metric system is base 10 (because of ten fingers) whereas computer works in binary (ON/OFF) hence kilo in binary is  $2^{10}$ , which is nearest to 1000, and equals 1024.

**[2 marks]**

4. For example:

Multi user is many users using the same computer system **[1 mark]** such as networked computers **[1 mark]** – accept an example of a networking O/S if it is clearly for networked systems (some candidates have read the question as asking for an O/S example)

Multi tasking for example a single user system where the user is working on one program **[1 mark]** whilst an anti virus program is checking the hard disk at the same time **[1 mark]** – do not accept Windows unless an example of multi tasking is given.

**[4 marks]**

5. Award **[1 mark]** for identifying type of transfer lines and **[2 marks]** for explaining the difference this makes. For example:

- Introduction of digital lines **[1 mark]**, lines that can hold both digital and analogue **[1 mark]** have speeded up the transfer and hence reduced the error. **[2 marks]**
- Fibre optic lines even faster and more accurate. **[2 marks]**
- When data is sent digitally it needs no conversion at either end. Less prone to error. **[2 marks]** **[3 marks max]**

*The answer must refer to the feature of communication lines. Accept such features as repeaters to boost the signal and well insulated lines.*

6. Award **[2 marks]** for the role of each of the components explained.

For example:

Program counter holds address of next instruction to be executed;  
 Bus (address bus) used to access address of appropriate instruction;  
 And take instruction to instruction register (data bus);  
 Instruction when fetched placed in instruction register;  
 Program counter incremented;  
 Instruction executed;  
 Cycle repeated;

**[6 marks max]**

7. (a) Award **[1 mark]** for correct method but incorrect answer.

01110010 shifted 0101;  
 = 011100.10;  
 = 28 ½;

**[3 marks]**

*If the exponent is not used in part (a) then no marks. However, be lenient for the rest of the question.*

- (b) 113 = 01110001 shifted 0111

= 0.1110001 / 0111 x 2

10001110 / 0111

+ 1

-113 10001111 / 0111

**[3 marks]**

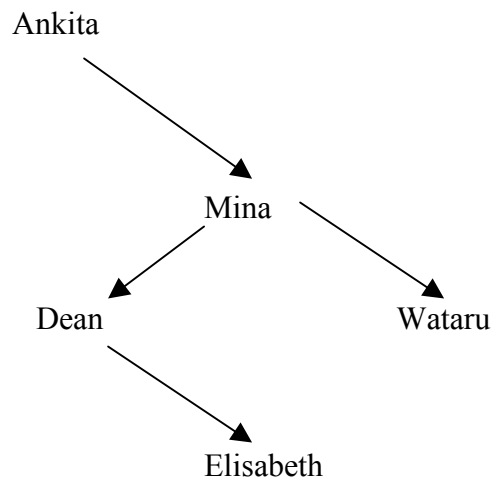
*Award **[1 mark]** for 113 converted to binary*

***[1 mark]** for 2's complement – can be done in one step without writing 113 -*

***[1 mark]** for the correct shift*

- (c) truncation error occurs when the number will not fit into the required register **[1 mark]**  
 least significant bits are discarded **[2 marks]**

8. Award **[1 mark]** for each name correctly placed after the root.



Accept the mirrored equivalent.

**[4 marks max]**

9. Award **[1 mark]** for each correct BigO and **[2 marks]** for an elaborate comparison.  
 Bubble sort is  $O(n^2)$  **[1 mark]** since it involves nested loops, whilst quicksort is  $O(n \log n)$  **[1 mark]**, where  $\log n$  is less than  $n$  hence is more efficient **[2 marks]** for comparison.

**[4 marks]**

10. Direct memory access does not involve the processor once transfer is started and is used, for example, for data transfer between disk drive and memory. **[2 marks]**

**Section B**

11. (a)	Number	return value
	5	
	4	
	3	
	2	120

*Award [2 marks] for reducing from 5 to 2 [2 marks] for correct value of return value.  
[4 marks]*

- (b)  $\text{FACT}(5) = 120$  from above  
 $\text{FACT}(2) = 2$   
 $\text{FACT}(5-2) = \text{FACT}(3) = 6$   
Hence  $\text{PERM}(5,2) = 120 / (2 \times 6) = 10$

*Allow [1 mark] for each of the above lines and do not penalise candidate for wrong value from first section. [4 marks]*

- (c) The function `FACT` changes the value of the number. If this is carried back to `PERM` then the evaluation of the expression will not be correct;  
An expression such as  $(n - r)$  can't be passed by reference/only by value. [2 marks]

Accept that `FACT` cannot be recalled with an expression  $(N - 1)$  if the variable is by reference.

*Award [1 mark] for stating that the values of  $n$  and  $r$  will change and [1 mark] for identifying the effect.*

12. (a)

A	B	C	X	Y
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	0

*Award [1 mark] for all inputs shown.*

*Award [1 mark] for the three lines where both outputs are 0*

*Allow [1 mark] for correct lines with  $X=1$  and  $Y=0$*

*Allow [1 mark] for correct lines with  $X=1$  and  $Y=1$*

**[4 marks max]**

(b)  $X = C + A \cdot \bar{B}$

$X = C$  or (A and not B)

$$Y = C \cdot (A \cdot \bar{B} + \bar{A} \cdot B)$$

$Y = C$  and ((A and not B) or (B and not A))

$Y = C$  and ((A and not B) or (B and not A)) gets full marks as does

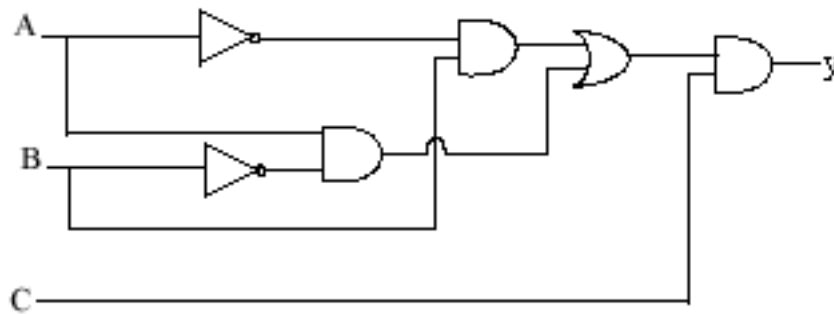
$Y = C$  and (A XOR B)

*Accept any symbols used as long as it is clear. Since there is only one mark do not penalise if the expression is not simplified.*

*Accept any correct expression that follows the truth table.*

**[2 marks max]**

(c)



*Allow [2 marks] for the NOT and AND gates applied to A and B, [1 mark] for correctly combining with an OR gate and [1 mark] for the final AND with C.*

**[4 marks]**

**N.B.** that XOR can be used instead of combinations of AND and NOT.



13. (a) *interrupts*  
 terminal/provider sends interrupt requiring service **[1 mark]**  
 flag set on server **[1 mark]**  
 if no higher priority input taken from terminal/internet **[1 mark]** **[2 marks max]**
- polling*  
 each user of system given a slice of time **[1 mark]**  
 by system going to each in turn **[1 mark]**  
 if input required then taken from user whose turn it is **[1 mark]** **[2 marks max]**
- in this case high number of users makes it time wasting to examine each **[1 mark]** since  
 at any time the demand is unpredictable **[1 mark]** especially with internet connection  
 which would not be feasible with polling **[1 mark]**. **[2 marks max]**  
**[6 marks max]**
- (b) *Allow [2 marks] for direct access and [2 marks] for indexed access. Also credit advantages and disadvantages of each.*

For example:

Records unordered;  
 Direct means a hashing algorithm;  
 Should be fast in this case;  
 Could be wasteful of disk space;

Partially indexed means finding block quickly;  
 Could involve lengthy sequential search in block;  
 Records ordered;

**[4 marks max]**

14. (a) Award **[1 mark]** for each of the following and only if the correct description given.

Position of vertices – data  
 Number of sides – data  
 Length of side – data  
 Resizing – operation  
 Colouring – operation  
 Rotating – operation

Award **[2 marks max]** for data members and **[2 marks max]** for methods. The candidate has to say which is which for full marks. If a list of data and methods is given (with 2 of each) then allow total **[2 marks]**.

**[4 marks max]**

- (b) No need to be concerned with the way in which the object is composed.  
 Data is hidden from the creator of the screen display who can create and place shapes without fear of changing the basic shapes. **[2 marks max]**

- (c) For example:

Allow **[2 marks]** for inheritance and **[2 marks]** for polymorphism.

Basic polygon shape defined

Particular shapes such as triangle *etc.* derived from polygon *inheriting* the characteristics and adding some of their own **[2 marks]**. **[2 marks max]**

*polymorphism*

When colouring (filling) or rotating *etc.* the result will vary as to the shape called *polymorphism* **[2 marks]**.

**Or**

If variables holding shapes are declared as different types (e.g. square, triangle) **[1 mark]** without *polymorphism* would need to call 'fill square' or 'fill triangle' **[1 mark]**.

**Or**

*polymorphism* allows a generic call **[1 mark]**

call then routed to the correct routine **[1 mark]**.

**[2 marks max]**

**[4 marks max]**

15. (a) *Award [1 mark] for advantages and disadvantages of static (compared with dynamic) and [1 mark] for advantages and disadvantages of dynamic (compared to static).*  
**[2 marks max]**

For example:

Static queue: simple to implement but restricted size may need reshuffling

Dynamic: algorithm complicated, could take time, but allows flexibility in length of queue  
**[2 marks]**

- (b) *Award [1 mark] for each of the following points illustrated or described.*

Start at head pointer in Q1 and traverse to the end, counting the number in queue;

Start from head again and traverse until third from end (length now known);

Traverse a pointer to end of Q2;

Reassign pointer from previous queue member to the second from last in Q1

Reassign rear pointer of last member in Q2 to moved member

Set pointer of moved member to null

**[6 marks]**

- (c) No need to traverse pointers twice to reach third from end of Q1;

List can be easily traversed from either end;

**[2 marks]**