

MARKSCHEME

May 2002

COMPUTER SCIENCE

Higher Level

Paper 1

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 penalised. 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. Allow **[1 mark]** for data type:

string;
array of character;
Accept numerical storage if justified (e.g. takes less space).

Any [2 marks] for explanation:

leading zeros;
spaces;
No need to do arithmetic, may need to search on code (1st 5 digits).

[3 marks]

2. Award **[1 mark]** for type of search and **[1 mark]** for reason.

type of search: linear/sequential search
for example: other searches require sorted data
consecutive data is not related
cannot be sure that the item searched is in the list until all items have been inspected.

[2 marks]

3. **[1 mark]** for each. Any **[3 marks]** for:

temporary file
used to save the results of individual transactions in *real time* when
access to the master needs to be restricted.
The data from the transaction file being used later to *update the master file in batch mode.*

[3 marks]

4. Allow **[1 mark]** for each item, up to a maximum of **[2 marks]**.

Item:
technical specification;
design drawings;
pseudo-code;
data dictionary;
purpose of algorithms;
contact details *etc.*

[1 mark] for each reason, up to a maximum of **[2 marks]**.

Reason:
so that several people can work on a project at the same time;
so projects can be maintained, updated / debugged *etc.* at a later date.

[4 marks]

5. *[1 mark] for correct answer and [1 mark] for correct working.*
10100101

[2 marks]

6. Yes.

[1 mark]

7. *Award [1 mark] for method, [1 mark] for description, and [1 mark] for method of failure avoidance.*

parallel: old and new system working simultaneously;
so that if there is a problem users can revert to old system;
until the bugs are sorted out;

modular or phased: bringing in one part of the system at a time;
checking that each is working correctly;
before introducing the next;

[3 marks]

Candidates who choose to backup the whole system and then revert to the backup if it all goes wrong have not explained a method of upgrading so should not get the first mark. Award as follows:

backup the whole system;
so that it can be reinstated if the new system fails;

8. *[1 mark] for each.*
both are reusable blocks of code which can be repeatedly called;
both handle parameters in the same way allowing pass by value and pass by reference *[2 marks]*. Allow *[1 mark]* for incomplete but correct comparison.
both declare local variables for use in function / procedure;
additionally functions return a value;
which is substituted in-line from the call;
function is declared with variable type; *[5 marks max]*
9. *[1 mark] for each.*
logical errors do not give expected result;
syntax errors break the rule of language grammar;
run-time errors produce situations program can't handle; *[3 marks]*
10. mantissa: number;
exponent: shift or power of 2; *[2 marks]*
Accept comparison with standard form.
11. the msb sets the sign;
a large number using the msb in its magnitude will appear to be negative. *[2 marks]*
12. *Award [1 mark] for any of the following.*
An interpreter would be used:
to translate and execute instructions;
stopping when error found;
often during development of program;
to correct errors without creating executable code;
create executable code when debugged. *[2 marks]*
13. *[1 mark] for each of the following.*
allows same operation;
on more than one object;
behaviour depends on the object;
simplifies coding / makes it generic; *[3 marks]*
Accept answers that refer to "operator overloading" or functions that accept different parameter types.
14. *Award [1 mark] for each if correct.*
 $O(n)$;
 $O(n^2)$;
 $O(n \log n)$. *[3 marks]*
15. *Award [1 mark] for any valid point.*
data may be wrong;
should be able to challenge it;
have the right to know what organisation has access to;
in order to prevent abuse. *[2 marks]*

SECTION B

16. (a) *[1 mark] for each correct line. First 2 lines given.*

I	C	D	A[]
197	0	128	1000 0000
69	1	64	1100 0000
5	2	32	1100 0000
5	3	16	1100 0000
5	4	8	1100 0000
5	5	4	1100 0100
1	6	2	1100 0100
1	7	1	1100 0101

[6 marks]

(b) stores the binary equivalent of I in A[].

[1 mark]

(c) convert I into a pair;
of hexadecimal magnitude values / whole number and remainder when divided
by 16;

[1 mark]

*Accept any answer that shows an understanding of **mod** and **div**.*

(d) convert A and B integers;
into hexadecimal characters.

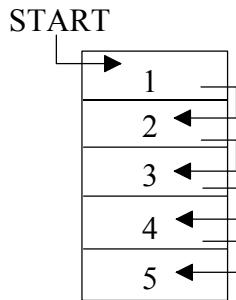
[1 mark]

(e) converts a short integer into a hexadecimal value.

[1 mark]

17. (a) each name may have more than one entry;
and primary keys must be unique. *[2 marks]*
- (b) to prevent duplicate entry;
and maintain data integrity. *[2 marks]*
- (c) hourly rate taken from table 1 using key;
combined key finds person + week in table 2;
hours worked returned;
Multiply by hourly rate to return wage. *[4 marks]*
- (d) Disadvantage
two keys could give the same storage address;
overflow takes time / space.
- Advantage
fast retrieval since data is found without searching; *[2 marks]*

18. (a) when the stack or queue is of a fixed size; and/or maximum size is known; **[1 mark]**
- (b) easier to program;
 therefore more robust;
 fewer problems with memory leakage / pointer manipulation;
Accept "faster access" if justified (e.g. reference to static / dynamic memory) **[1 mark]**
- (c) can cope with structures of indeterminate maximum size; **[1 mark]**
- (d) **[1 mark]** for any of the following.
 doubly linked lists are used for forward and backward chaining;
 this is never needed in stacks and queues;
 because the data is always handled at the front or back; **[2 marks max]**
- (e) **[1 mark]** for any of the following.
 circular lists are used in static structures to minimize overflow;
 as linked lists have no limit;
 there is no need for a circular structure;
 no need to have tail pointer to head; **[2 marks max]**
- (f) **[1 mark]** each for correct addition and deletion.
[1 mark] for in a stack situation.
 For example:



Delete: change start pointer to point to pointer 2;

Add: change start pointer to new item pointer from new item to item 1;

[3 marks]

19. (a) Award **[1 mark]** for all combinations of A, B, C present and **[1 mark]** for each pair of corresponding H and L.

A	B	C	H	L
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- (b) Full adder;
Allow **[1 mark]** for adder.

[5 marks]
[2 marks]

- (c) **[1 mark]** for $AB + BC + AC$;
[2 marks] for correct working either Karnaugh Map or Algebra.

[3 marks]

20. (a) Advantage: commands quickly typed and translated to instruction;
Disadvantage: need to know the instructions home not user friendly; **[2 marks]**
- (b) Advantage: cannot make mistakes;
Disadvantage: not flexible, restricted to combination available; **[2 marks]**
- (c) Advantage: do not need to be an expert to use it;
Disadvantage: uses memory; **[2 marks]**
- (d) *Credit any of the following with [1 mark].*
job or device needs attention;
interrupt sent;
using flag;
processor suspends current job;
attends to job or device;
flag reset to 0;
resumes current job;
interrupts are handled in order of priority/relative importance; **[4 marks]**
-