IBO IB DIPLOMA PROGRAMME
PROGRAMME DU DIPLÔME DU BI
PROGRAMA DEL DIPLOMA DEL BI

N05/5/COMSC/HP2/ENG/TZ0/XX/M+

MARKSCHEME

November 2005

COMPUTER SCIENCE

Higher Level

Paper 2

12 pages

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

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) by telephone. 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 Team Leader by telephone. 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. Follow the markscheme provided, do **not** use decimals or fractions and mark only in **RED**.

2. 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.

3. 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.

4. Unexplained symbols or personal codes/notations on their own are unacceptable.

5. 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.

6. Where an answer to a part question is worth no marks, put a zero in the right-hand margin.

7. Record the mark awarded for each of the five questions answered in the Examiner Column on the cover Sheet.
Add up the marks awarded and enter this in the box marked TOTAL in the Examiner Column on the cover sheet.

8. 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.**

9. 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.

10. 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 2 Markscheme

Mark Allocation

Candidates are required to answer ALL questions (***[30 marks]*** for question 1, ***[30 marks]*** for question 2 and ***[15 marks]*** for the remaining three questions). Maximum total = ***[105 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"**.

1. (a) *Award **[1 mark]** for each of the following*:
definition or ID/key field;
string for name;
unsigned integer for size; ***[3 marks]***

(b) For example:

```
procedure selectsort
    declare i, j, min integer
    declare temp datatypefrompreviousanswer
    for i <--0 to 99 do
       if TRANSACTION[i].FILENAME = nil then
           i <-- 99
       else
         min <-- i
         for j <--i to 99 do
           if TRANSACTION[j].FILESIZE = nil then
                  j <-- 99
           else if TRANSACTION[j].FILESIZE<TRANSACTION[min].FILESIZE then
                  min <-- j
           endif
         endfor
         temp <-- TRANSACTION(min)
         TRANSACTION(min) <-- TRANSACTION(i)
         TRANSACTION(i) <-- temp
       endif
    endfor
endprocedure selectsort
```

There maybe some differences but the sort must be a selection sort.
Award marks as follows:

declaration of variable	1
correct outer loop	1
test for nil filename and action	2
set minimum place value	1
correct inner loop	1
test for nil filename and action	2
correct comparison and min reset	2
values swapped correctly	2

[11 marks max]

(c) For example:

```
procedure merge
    declare LST, MAS, TRA, TMP integer
    declare OK boolean
    MAS = TRA  = LST = 0
    repeat
       LST = LST + 1
    until master[LST] = nil
    OK <--  true
    while OK
       if master[MAS] > transaction[TRA] then
           TMP <-- LST
           repeat
               master[TMP] <-- master[TMP-1]
               TMP <-- TMP - 1
           until TMP = MAS
           master[MAS] = transaction[TRA]
           LST = LST + 1
           TRA = TRA + 1
           if transaction[TRA] = nil or LST = 200 then
           OK = false
           endif
        endif
        MAS = MAS + 1
    endwhile
endprocedure merge
```

Award ***[1 mark]*** *for each of the following*:
Declaration of variables;
Initialisation of variables;
Check through MASTER until nil to determine size;
Set up loop to continue till end of merge;
Compare MASTER and TRANSACTION filesize;
If TRANSACTION smaller;
Shuffle down MASTER records;
Write TRANSACTION record to MASTER;
Adjust position in TRANSACTION file;
Check for end of TRANSACTION file;
If end exit check for MASTER fault;
Adjust position in MASTER file;
Check for end of MASTER file;
Write rest of TRANSACTION to MASTER if so; ***[12 marks max]***

This is a difficult algorithm but the above scheme should allow candidates to gain credit for a reasonable attempt. Accept writing to a new master file.

(d)
- If TRANSACTION is empty;
- write nil values to all records;
- else shuffle all remaining records to the front of the queue;
- and write nil values from the end of the new queue to the end of the array; ***[4 marks]***

2. (a) *Award* ***[2 marks]*** *for employers advantages and* ***[2 marks]*** *for employee advantages.*

Employers
Employers need fewer workers;
and therefore reduce costs;
employers can give skilled workers more responsibility;
and therefore improve performance;

Employees
Employees can work fewer hours;
and therefore have time for development;
or holidays; ***[4 marks max]***

(b) Although costs are measurable;
benefits are less measurable;
for example corporate image (or any other reasonable example);
can also improve the overall performance of a business; ***[4 marks]***

(c) *Award* ***[1 mark]*** *for each of the following up to* ***[2 marks max]****.*

Complex systems can develop emergent properties;
which, because they are unpredictable;
cannot be planned for; ***[2 marks max]***

(d) *Award* ***[1 mark]*** *for any of the following up to* ***[4 marks max]****.*

Businesses who hold information about people need to ensure that it is not used for the wrong purpose;
Or hacked by someone else;
When information is held electronically it is easier for it to be used without anyone knowing that it has been tampered with;
Who does the information belong to?;
What about the ease of plagiarism?;
etc. ***[4 marks max]***

(e) (i)

1. Problem
2. Design / Analysis
3. Implement
4. Evaluate
5. Goto 1 with new problem ***[3 marks]***

Award ***[2 marks]*** *for the four stages and* ***[1 mark]*** *for the cycling back to the beginning.*

(ii) prototype has the same cycle back to beginning ***[2 marks]***
but also includes a second inner cycle at the design stage;
where user tests the prototype;
which generates feedback;
and loops until satisfied; ***[5 marks max]***

(f) A prototype is designed to elicit user information;
It is not designed as an end product;
The final design should not be developed from the prototype because its design may not be optimal for a final design;
To start from scratch enables final design criteria to be incorporated; ***[4 marks]***

(g) *Award* ***[2 marks]*** *for any two of the following.*
End user development ***[1 mark]*** is risky ***[1 mark]***
Off the shelf ***[1 mark]*** may not mesh ***[1 mark]***
Outsourcing ***[1 mark]*** relinquishes control ***[1 mark]*** ***[4 marks max]***

3. (a) To find a memory block the read/write head ***[1 mark]*** has to locate the correct track ***[1 mark]*** this is called the seek time ***[1 mark]*** then the head has to wait until the target block rotates around to it ***[1 mark]*** this is the latency. Then the data has to be read or written. The sum of all these times is the access time ***[1 mark]***. ***[5 marks]***

(b) (i) *Award* ***[1 mark]*** *for each of the following*:
As a disk is used and fills up, data may not be stored in contiguous blocks;
because a contiguous space is not available;
therefore instead of accessing one contiguous location the disk has to access *several*;
thus "increasing" the access time; ***[2 marks max]***

(ii) *Award* ***[1 mark]*** *for each of the following*:
Disk is scanned for files which are not stored in contiguous blocks;
Data rearranged so that different parts of the same file are stored together;
Empty space caused by previously deleted files used bring blocks together;
[2 marks max]

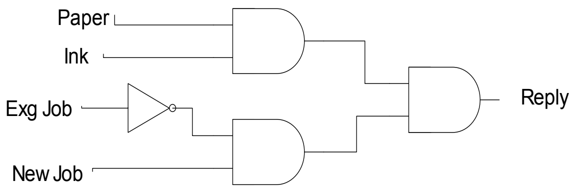
(c) (i) *Award* ***[1 mark]*** *for each of the following*:
RAM is accessed directly;
via address buses;
and data buses; ***[2 marks]***

(ii) *Award* ***[1 mark]*** *for each of the following*:
disk has moving parts;
RAM has no mechanical movement involved;
hence faster; ***[2 marks max]***

(d) *Award* ***[1 mark]*** *for each of the following*:
Interrupt register holds flag for each call to interrupt the CPU;
each cycle the CPU examines register and calls highest priority position flag to react to;
[2 marks]

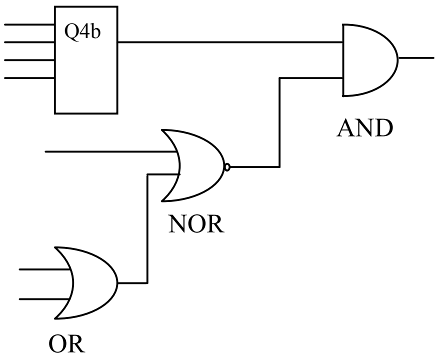
4. (a) Handshaking; ***[1 mark]***

(b) *Award **[1 mark]** for each gate connected correctly and **[1 mark]** for all symbols correct.*



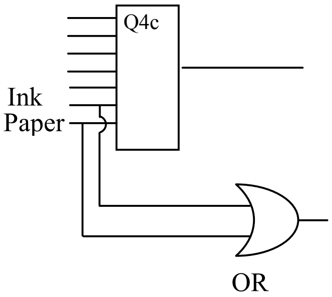
[5 marks]

(c) *Award **[1 mark]** for each correctly drawn and connected gate.*



[3 marks]

(d) *Award **[1 mark]** for connections and **[1 mark]** for correct symbol.*



[2 marks]

(e) *Award **[1 mark]** for advantage and **[1 mark]** outline.*

For example:
Can be re-programmed ***[1 mark]*** so if changes are needed no need to make new circuit ***[1 mark]***.

*Award **[1 mark]** for disadvantage and **[1 mark]** outline*

For example:
If speed is important a circuit can react more quickly ***[1 mark]*** than a program which is executing instructions ***[1 mark]***. ***[4 marks]***

5. (a) Test data would be selected;
to test for normal;
abnormal;
extreme;
If the program does not react to all data correctly the relevant code should be redesigned and re-tested; ***[5 marks]***

(b) Testing should be planned as soon as possible after the user requirements have been identified ***[1 mark]*** so that design and implementation do not influence it ***[1 mark]*** and testing should address requirements as well as design ***[1 mark]***. ***[3 marks]***

(c) Analyst for same reasons as Q5b. ***[1 mark]***
Both for ***[1 mark]***.

(d) *Award* ***[2 marks]*** *for validation and* ***[2 marks]*** *for verification.*

For example:

Validation checks to see if a data item is reasonable or not. ***[1 mark]***
e.g. the grade should be in the range 0 – 100. ***[1 mark]***

Verification checks to see if an item of data is really what the user wants ***[1 mark]*** *e.g.* double entry of the grade ***[1 mark]***. ***[4 marks]***

(e) Validation should come first ***[1 mark]*** because if it fails verification will not be required ***[1 mark]***. ***[2 marks]***