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

MARKSCHEME

May 2005

COMPUTER SCIENCE

Standard Level

Paper 2

12 pages

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

Subject Details: Computer Science SL Paper 2 Markscheme

Mark Allocation

Candidates are required to answer ALL questions. (***[30 marks]*** for question 1, ***[25 marks]*** for question 2 and ***[15 marks]*** for question 3.) Maximum total = ***[70 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 any of*:

it is a sentinel/end-of-data marker/rogue value;
that marks the end of the list of names;
so the programmer knows how many names/students there are; ***[1 mark]***

(b) A possible solution is:

```
procedure ABSENCES ( val ATTENDANCE array of Boolean(1..N, 1..5),
                     val NAMES array of String(1 .. N) )
/*
  Algorithm to output ATTENDANCE array data for a given name
*/
   declare ROW, COL integer
   declare NAME String
   declare FOUND, LISTEND Boolean

   output "Input student name "
   input NAME

   ROW = 1
   FOUND = FALSE
   LISTEND = FALSE

   // find correct ROW in NAMES array

   while ( (not FOUND) and (not LISTEND) ) do

      // check for end of list
      LISTEND <-- (NAMES[ROW] = "ZZZ")

      if not LISTEND then
            FOUND <-- (NAMES[ROW] = NAME)
            if not FOUND then
               ROW <-- ROW + 1
            endif
      endif
   enddo

// output absence data

   if FOUND then
      for COL <-- 1 upto 5 do
            if not ATTENDANCE[ROW, COL] then
               output "absent for day ", COL
            endif
      endfor
   else
      output("No such student in class");
   endif

endprocedure ABSENCES
```

*Award marks as follows up to **[10 marks max]** (out of 11 marking points)*

Award ***[1 mark]*** *for each of the following*:
declaring most local variables used;
input of name/use of string parameter for name;

linear search algorithm ***[4 marks max]*** for the following:
correct initialization;
loop through values until found;
check for end of list marker (`ZZZ`);
correct exit value for `ROW` or equivalent;
Allow this mark for a nested 2nd loop.

Award ***[1 mark]*** *for*:
test for `NAME` found;

loop through `ATTENDANCE` ***[3 marks max]*** *for the following*:
using `ROW` from search (allow follow through marks if `ROW` incorrect from search);
loop through all 5 columns;
suitable output message (acceptable if candidate has simply output an absence total);

Award ***[1 mark]*** *for*:
output of error message *"not found"* in correct branch of test; ***[10 marks max]***

(c) A possible solution is:

```
procedure INSERT ( val STUDENT string,
                   ref NAMES array of String(1 .. N) )
/*
  Algorithm to insert STUDENT into NAMES array
*/
  declare POS integer    // end of name list
  declare INS integer    // position to insert
  declare P integer      // shuffle pointer

  /*  search up the array to ZZZ
      Make a note of where the insertion position is */

    POS <-- 1;
    INS <-- 1;

    while NAMES[POS] # "ZZZ"

      if NAMES[POS] < STUDENT then
        INS <-- INS + 1
      endif

      POS <-- POS + 1

    endwhile

      /* shuffle down to insertion position */

    for P <-- (POS + 1) downto INS
      NAMES(P) <-- NAMES(P - 1)
    endfor

      NAMES[INS] <-- STUDENT

endprocedure INSERT
```

*Award marks as follows to a maximum of **[10 marks]**.*

*Award **[1 mark]** for each of*:
NAMES array as a reference parameter;
declaring all local variables used;
initializing local variables appropriately;

Locating the end of the array marker position (ZZZ) ***[2 marks]***.
*Award **[1 mark]** for a slightly incorrect attempt.*
Allow these marks to candidates who simply shuffle the entire array from N down to the correct insertion position.

Locating the correct insertion position ***[2 marks]***.
*Award **[1 mark]** for a slightly incorrect attempt.*

*Award **[1 mark]** for*:
Shuffle down:
correct start point for shuffle loop;
correct end point for shuffle loop;
correct shuffle;
insertion of name in correct location; ***[10 marks max]***

(d) *Award **[1 mark]** for any of the following up to **[2 marks]**.*

Attendance data has not moved with correct student;
Student marked as absent when not; ***[2 marks]***

(e) *Award **[1 mark]** for each of the following up to **[5 marks max]**.*
search through `NAMES` to count number of students, n for each column (for 1 to 5);
loop from 1 to n;
count when not true (or false);
compare which is the greatest;
output the day corresponding to the largest;
monday is column 1;
etc. ***[5 marks max]***

(f) *Award **[1 mark]** for identifying a limitation a further **[1 mark]** for additional elaboration.*

Fixed dimensions; could run out of space to add more names; could run out of space to add more days;
Limited data type in `ATTENDANCE`– only 2 entries possible, but may need to mark, *e.g.* holidays as neither absent nor present;
Inefficient storage; most entries are true; could record only name, date of absence;
[2 marks max]
Sorting one of the arrays;
Requires sorting the other; to maintain accuracy; array of objects/records is better; because name and attendance are kept together;
No unique ID is given for a student; two students could have the same name;

2. (a) (i) *Award **[1 mark]** for each of the following points, up to **[2 marks max]**.*

a protocol is a set of rules;
by which devices on the network communicate;
such as speed, parity checks (any suitable answer); ***[2 marks max]***

(ii) Award ***[1 mark]*** for identifying an issue, up to 2 for further elaboration 2 x 3 issues up to ***[6 marks]***. The issue can be implicit.

Examples of issues include:

Compatibility; the company should select equipment/applications which work on existing protocols; to simplify installation;

Conversion requirements; the company will need to think about software/hardware for converting data among different protocols when new equipment is installed;

External communications; the company should think about systems it needs to take data from or to pass it to; to ensure same protocols are used/converted;
Security; only equipment/computers with correct protocol/encryption; can access network; ***[6 marks]***

(b) *Award **[1 mark]** for a way in which they could be used related to the case study and **[1 mark]** for further elaboration* x *2 to* ***[4 marks max]***. *For example*:

financial modelling;
projections of expenses, revenues, profits *etc*;

decision support;
construction of what-if scenarios (to investigate different possible outcomes);

network traffic;
to eliminate potential bottlenecks in the system; ***[4 marks max]***
Any other reasonable suggestion should be accepted.

(c) *Award **[1 mark]*** for an identified problem up to ***[2 marks]*** for further explanation x 2 to ***[6 marks max]***. *For examples*:

Sales of restricted goods;
Not available locally (by law);
Can be ordered via website;

Issues which solely relate to LAM's and individuals are not acceptable – e.g. employee monitoring, hacking, spamming.

crime;
which country has jurisdiction in computer crime;
where the server is located;
the management head office;
where the victim is located;

ownership;
e-business websites, educational resources; owned by creators or employers;
ownership of ideas/intellectual property;

taxation;
who pays;
where do they pay;
what benefits do they expect;
etc. ***[6 marks max]***

(d) *Award* ***[1 mark]*** *for each of the following up to* ***[3 marks max]****.*
Answers must relate to "early" design
increased cost of late changes ('cost of fix at design time is multiplied one thousand fold at post release time');
reduced risk of project failure;
increased user involvement increases likelihood of solving the "real" problem;
leading to increased user acceptance;
more reliable systems are designed;
more time is devoted to potential problems;
solutions are more effective; ***[3 marks max]***

(e) *Award* ***[1 mark]*** *for an advantage and* ***[1 mark]*** *for further elaboration.*

For example:

staff can train on new system; while old one is running;
can check that same results are achieved;
can see that nothing goes wrong;
if they make a mistake, old system still gives correct results;

Award ***[1 mark]*** *for a disadvantage and* ***[1 mark]*** *for further elaboration.*

For example:

staff have more work; to maintain both systems;
new staff maybe introduced leading to redundancies of existing staff;
increased cost; ***[4 marks max]***

3. (a) *Award up to **[4 marks]** for each of the following*:

ease of use; allows fast data entry;
easier to use than text-based system;
via use of images/icons; pointing system; menus;
screen space restricted; picture more compact than text;
easier to train users; ***[4 marks max]***

(b) *Award **[1 mark]** for each of the following up to **[5 marks max]**.*

id/key field and also the meter reading are in the handheld;
they are transferred to the computer system/a transaction file;
This key is found in the master file (at the company office);
It is used to locate the customer record which contains;
fields such as id, name, address, last meter reading, *etc*;
the new meter reading is used to calculate the bill;
which is sent to the customer using data from the file;
the master file is updated; ***[5 marks max]***

(c) *Award **[1 mark]** for a device, one for a description* x *2 up to **[4 marks max]**.*

keypad; used to enter numeric/text data;
pen; to enter data (or signatures) by handwriting;
pointing device; to access menus/icons;
any automatic data entry method (*e.g.* via a communication port);
any direct data entry method (*e.g.* OCR) however bizarre it may seem;
speech recognition; data input via microphone; ***[4 marks max]***

(d) *Award **[1 mark]** for each identified advantage and disadvantage TO THE METER READER, **[2 marks max]**.*

For examples:

Quicker to access data from handheld (not to transfer data at office);
Less likely to make data entry errors (as can be validated);
May lose battery power;
no backup system for data recording;
volume of data stored (in small space);
do not accept issues of portability or damage; ***[2 marks max]***