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

MARKSCHEME

May 2007

COMPUTER SCIENCE

Standard Level

Paper 2

9 pages

Subject Details: Computer Science SL Paper 2 Markscheme

Mark Allocation

Candidates are required to answer ALL questions. (***[20 marks]*** for question 1, ***[20 marks]*** for question 2, ***[30 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) the original ticket contains a bar-code/magnetic strip/MICR etc;
on which is coded the time of entry;
this can be read by the pay-machine; ***[3 marks]***

(b)
```
public double charges (int hours)
{
     return (3 + (hours – 1) * 2.50);
}
```
Award marks as follows:
[1 mark] *for the word* ***double*** *in the method signature.*
[2 marks] *for the correct calculation (1 for a good but incorrect effort).*
[1 mark] *for returning a value.* ***[4 marks]***

(c) America; ***[1 mark]***

(d)
```
public int time (String start, String finish)
{
  String startHour, finishHour, startMin, finishMin;
  int hour1, hour2, min1, min2;

  startHour  = start.substring(0,2);
  finishHour = finish.substring(0,2);
  startMin   = start.substring(3,5);
  finishMin  = finish.substring(3,5);

  hour1  = Integer.parseInt(startHour);
  hour2  = Integer.parseInt(finishHour);
  min1   = Integer.parseInt(startMin);
  min2   = Integer.parseInt(finishMin);

  if (min2 > min1)
       return ((hour2 – hour1) + 1);
    else  return (hour2 – hour1);
}
```
Award marks as follows:
[1 mark] *for the type* ***int*** *in the method signature.*
[1 mark] *for correct declaration of variables.*
[2 marks] *for correct substring conversion with* ***[1 mark]*** *for a good but incorrect attempt/or a consistent error with the indexing.*
[1 mark] *for correct parsing.*
[2 marks] *for a correct calculation with* ***[1 mark]*** *for a good but incorrect attempt.*
[1 mark] *for returning a value.* ***[8 marks]***
There are alternate solutions, e.g. turning both times into the number of minutes since midnight, finding the difference, dividing by 6 etc. The same markscheme should be able to be used for such solutions.

(e) (i) the car-park could have a car entering one day and leaving the next;
this would give an incorrect number of hours in the function `time()` / incorrect cost;
[2 marks]

(ii) No. of hours = (24:00 – entry time) + (departure time - 00:00) / or equivalent explanation;
then add 24 to total for each subsequent day parked / or equivalent explanation; ***[2 marks]***

2. (a) (i) tests if the value of `x` is `<` `y` / if the condition is true;
if it is / if (it returns) true, the code within the loop will be processed / loop will continue;
otherwise / if returns false, control will pass to the code following the loop structure / loop will end; ***[3 marks max]***

(ii)
```
public void multiples(int a, int y)
{
     int x = a;
     if (x < y)
     {
          do
          {
               System.out.println(x);
               x = x + a;
          }
          while (x < y);
     }
}
```
Award marks as follows:
correct initialisation for x;
correct if statement;
incrementing x;
correct do…while loop and condition; ***[4 mark max]***

(iii) the loop might never end / infinite loop / run time error;
because the value of x might 'jump' over the value of y / x might never equal y;
any correct scenario (e.g. y is not a multiple of a); ***[3 marks max]***

(b) (i) primary: any value from 2 Gbytes – 64 Gbytes;
cache: any value from 512 Kbytes – 4 Mbytes; ***[2 marks]***

(ii) code that is regularly used;
can be placed in the cache memory;
which is faster than normal memory;
saves time fetching code from normal memory; ***[3 marks max]***

(c) (i) expands the amount of RAM;
by using a section of secondary memory/backing store;
which allows the running of larger programs (than would normally be possible);
[3 marks max]

(ii) because if the RAM is small (relative to the size of the current program);
time will be lost with excessive memory swapping / transferring data from (the 'slower') virtual memory; ***[2 marks]***

3. (a) text is scanned into a (image) file;
OCR software converts into a readable format;
speech synthesis/screen magnification enables student to hear/read notes; ***[3 marks]***

(b) (i) Video Magnifiers;
cameras relay text to screen (where it can be magnified); ***[2 marks]***

(ii) Electronic Braille Displays;
enables the students to read the contents of the computer screen; ***[2 marks]***

(c) a working model of the interface is created;
a user is asked to trial it;
changes are suggested (by the user);
this cycle is continued as required; ***[3 marks max]***

(d) *Award **[2 marks]** for each feature up to **[4 marks max]**.*
any documentation should be adapted / printed in large type / Braille ***[1 mark]*** so that the users are able to read it ***[1 mark]***.
the workshop layout should be designed ***[1 mark]*** so that it is uncluttered / plenty of space in which to move etc ***[1 mark]***.
access to the workshop ***[1 mark]*** should not have steps etc ***[1 mark]***. ***[4 mark max]***

(e) areas that could be discussed are:
researching information;
communicating;
producing written work;

*For each area award **[3 marks max]** as follows*:

*Award **[1 mark]** for identifying the area.*
*Award **[1 mark]** for indicating the problem, and **[1 mark]** for a good expansion.*

For example:

Students will experience difficulties in researching information ***[1 mark]***. An invaluable source of information for projects etc., is the Internet ***[1 mark]***, access to which would prove difficult for these students without specialised equipment ***[1 mark]***. ***[6 marks max]***

(f) a Braille keypad might be useful for certain disabled students;
but would be difficult/impossible to operate for others;
whereas a voice recognition system would aid students with certain disabilities;
and would be able to be used by users who are not disabled; ***[4 marks]***

(g) *Award* ***[1 mark]*** *for a way,* ***[1 mark]*** *for an elaboration* ***[2 marks]*** *per example* × *3 up to* ***[6 marks max]***.

For example:

A wheelchair bound person;
could give commands via voice;

A physically-disabled/blind person;
could operate light switches/domestic appliances/their environment;

A physically-disabled/blind person;
could use it to enter text (into a word processor)/write a letter; ***[6 marks max]***