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

N06/5/COMSC/HP2/ENG/TZ0/XX/M

MARKSCHEME

November 2006

COMPUTER SCIENCE

Higher Level

Paper 2

15 pages

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). 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 TL and IBCA. 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. Once markscheme is received mark in pencil until final markscheme is received.

2. Follow the markscheme provided, do **not** use decimals or fractions and mark only in **RED**.

3. 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.

4. 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.

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

6. 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.

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

8. Record the mark awarded for each of the four 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.

9. 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.**

10. 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.

11. 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 (***[20 marks]*** for question 1, ***[20 marks]*** for question 2, ***[20 marks]*** for question 3 and ***[40 marks]*** for question 4. Maximum total = ***[100 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.
- 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"**.

*Answer **all** the questions.*

1. (a) Only integer values are stored;
values are small/don't exceed 255; ***[2 marks]***

(b) A possible solution is:

```
public int sum(byte[][] dice)
{
int sum = 0;
  for(int r = 0; r < 100; r++)
  {
      for(int c = 0; c < 3; c++)
      {
          sum = sum + dice[r][c];
      }
  }
  return sum;
}
```

Award marks as follows.

correct return type;
Accept byte anywhere in the method – even if the sums do get too big!
initializing sum;
using nested loops;
correct loop iteration on both loops;
correct summing of dice elements; ***[4 marks max]***

(c) A possible solution is:

```
public void evaluate
{
  for(int r = 0; r < 100; r++)
  {
      output("result, row: " + r + ": " + getPair(dice, r));
  }
}
```

Award marks as follows.

counter declared;
correct loop iteration (100 times);
correctly formatted output (permit minor errors like not putting a colon);
correct call to `getPair(dice, r)`; ***[4 marks]***

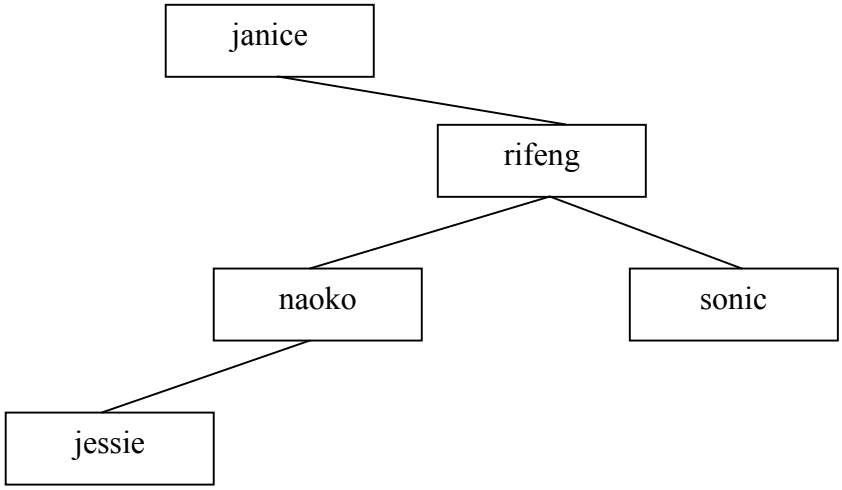
(d) A possible solution is:

```
private void createList()
{
  for (int x = 0; x < 100; x++)
  {
    // get row value and add to list
    byte val = getPair(dice, x);
    if (root == null)
    {
      // empty list add at front:
      DiceNode temp = new DiceNode();
      temp.setValue(val);
temp.setFrequency(1);
      root = temp;
    }
    else
    {
      // traverse to required node (or end of list
      DiceNode temp = root;
      DiceNode prev = root;
      while ( (temp.getValue() < val) && (temp.getNext() != null) )
      {
        prev = temp;           // store this for linking
        temp = temp.getNext();
      }
      // check node already exists
      if (temp.getValue() == val)
      {
        // exists - increment the frequency field
        temp.setFrequency(temp.getFrequency() + 1);
      }
      else
      {
        // add a new node here
        DiceNode newOne = new DiceNode();
        newOne.setValue(val);
        newOne.setFrequency(1);
      if (prev==null)
        {temp.setNext(newOne);}
      }
      else
      {
        newOne.setNext(prev.getNext());
        prev.setNext(newOne);
      }
    }
  }
}
```

Award marks as follows.

declaring most local identifiers used;
loop to one hundred;
call to `getPair(dice);`
testing root is null;
adding to head after test (or attempt at test); *Award* ***[2 marks]*** *if correct,* ***[1 mark]*** *for good attempt*.
walk down list; *Award* ***[2 marks]*** *if correct,* ***[1 mark]*** *for good attempt*.
checking node already exists;
incrementing if so;
inserting node;
inserting at end of list if appropriate; *Award* ***[2 marks]*** *if correct,* ***[1 mark]*** *for good attempt*.
[10 marks max]

2. (a)



[5 marks]

*Award **[1 mark]** for each node in a correct relative position.*
Accept the mirror image.

(b) A possible solution is:

```
public void insert(String[] names, String name)
{
  int pos = size; // start at top
  // shuffle 'til insertion point or first element
  while ( (pos != 0) && (names[pos-1].compareTo(name) > 0))
  {
    names[pos] = names[pos - 1];
    pos = pos - 1;
  }
  // insert here
  names[pos] = name;
  size = size + 1;
}
```

Award marks as follows.

declaring a local pos or equivalent;
assigning to size (or setting to zero initially);
attempt to loop through array until;
pos = = size or pos = = 0 (depending on direction);
insertion position located using compareTo();
attempt to shuffle in names array;
correct shuffle in names array;
attempt to insert at pos or similar;
incrementing size; ***[8 marks max]***

(c) Possible solutions are:

```
public int binaryItSearch(String[] names, int size, String wanted)
{
    int mid;
    int top = size;
    int bottom = 0;
    int found;

    while (top >= bottom)
    {
       mid = (top + bottom) / 2;
       found = names[mid].compareTo(wanted);
       if (found == 0)
       {
           return mid;
       }
       else if (found > 0)
       {
          // search further down
          top = mid - 1;
       }
       else
       {
           bottom = mid + 1;
       }
    }
    return -1;
}
```

Award marks as follows.
Since a fragment is asked for there is no need for the header, although it has been inserted.

all local identfiers declared and inititialised;
loop with correct termination condition;
calculation of mid or equivalent;
correct comparison (allow < and >);
use of compareTo;
Testing found or equivalent;
returning correct position, if found;
correctly moving top or bottom, if appropriate;
returning -1 or other suitable flag if not found; ***[7 marks max]***

3 (a) *Award up to **[2 marks]** for each technique that is described and an additional **[2 marks]** for a valid comparison*.

For example:

An index file could be used;
The correct logical order of usernames is specified in this file ***[2 marks]***.
Compared to an additional field, this doesn't require any changes to the existing file ***[2 marks]***.

A number field could be placed before each username;
This indicates its logical sequence ***[2 marks]***.
The first does not require a change to the record structure whereas the second adds an extra field/entry ***[2 marks]***.

Candidates may choose to read the file into a dynamic structure such as a binary tree or linked list storing details in the node sorted by username. Allow these and accept a valid comparison. ***[6 marks max]***

(b) *Award **[1 mark]** for each method/identifier and **[1 mark]** for a way in which it could be used,* x *3*.

For example:

identifier length/method length();
could be used to check the length of the username;

toLowerCase() method;
could be used to force a username to lower case;
could be used to compare username to a lowercase copy;

substring() method;
could be used to examine each char in turn;

charAt() method;
could used to examine each char in turn;
etc. ***[6 marks max]***

(c) *Award **[2 marks]** for the method and **[2 marks]** for identifying a problem.*

Method
the ASCII/UNICODE number of each character can be found;
these can be summed;
to give a number;
to which a modulo operator can be applied;
the result could correspond to a position in the file; ***[2 marks max]***
Accept alternate methods that produce a single number from the username.

Problem
two names can give the same result;
as there is a limited range of ASCII codes to sum; ('a' through 'z')
this is known as a collision/clash; ***[2 marks max]*** ***[4 marks max]***

(d) *Award marks as follows.*

(i) a number of characters are allocated to each field;
each field and the record have a pre-determined length;
there may be wasted space within each record;
it's easy to calculate the start position of a record;
direct (or sequential) access can be used; ***[2 marks max]***

(ii) each field is stored;
the data is terminated with a special character;
which can't be part of the data field;
the start of a particular record can't be determined by calculation;
full index or hash table cannot be used to access a record;
sequential access has to be used; ***[2 marks max]***

Technically an index could hold the position of the starting byte to seek to but candidates would have to explain this clearly to get a mark.

This question requires the use of the Case Study.

4. (a) *Award **[2 marks]** for outlining the process of handshaking and **[1 mark]** for explaining why MIDI data can be sent without handshaking*.

For example:

sending device sends a signal requesting to send;
receiving device returns ready signal;
data can be sent; ***[2 marks max]***

MIDI data sent uni directional and hence receiving device is always ready; ***[3 marks max]***

(b) (i) 1GB/10MB is 1024/10 so about 100 minutes. ***[2 marks]***

(ii) This will be 1024 times more. *i.e*. approx 100 000 minutes; ***[1 mark]***

(c) *Award **[1 mark]** for each comparison and **[1 mark]** for an elaboration, up to **[2 marks max]**, no more than 3 comparison*.

events not samples are stored;
musician's individual actions are recorded;
instruments/notes can be removed;
this is not possible with digital audio;

less storage required;
since audio recording has to sample "everything";
e.g. including pauses;
whereas MIDI does not record if nothing is being played;
MIDI takes 10Kbytes as compared to 10Mbytes for digital audio;

MIDI cannot handle singing;
which is possible with digital audio since it stores/samples "everything"; ***[6 marks max]***

(d) *Award **[3 marks]** for comparison and **[1 mark]** for why suitable*.
serial transmission involves sending data to one device then from that device to the next *etc*;
parallel transmission would involve sending the data to all MIDI devices at the same time;
serial means notes played one after the other;
parallel would mean all played together;
the time lag in serial is so small;
that the human ear cannot tell the difference; ***[4 marks max]***

(e) *Award **[2 marks max]** for parameters and **[4 marks max]** for process.*

parameters
*Award **[2 marks]** for all four of the parameters identified, **[1 mark]** for only three and no marks for one or two only.*
pitch
volume
attack
decay ***[2 marks max]***

process
Award marks as follows.
sound wave applied to give note ***[2 marks]***
this could be via frequency modulation synthesis – using two or more periodic signals ***[2 marks]***
or by wavetable synthesis – via a lookup table of sampled sounds ***[2 marks]*** ***[6 marks max]***

(f) *Accept any format if the reason is justified. Award **[1 mark]** for a sensible format, **[3 marks]** for a justification.*

For example:

Probably format 0;
- smallest format of file;
- best for transmission;
- despite lack of quality due to size reduction;
- there should be a compensation in the lack of time lag; ***[4 marks max]***

(g) *Award **[2 marks]** for each elaborated implication.*

For example:

digital music is easy to copy;
unauthorized "free" copies can be made and distributed;

there is no physical evidence of theft;
when a digital copy is made;

it is very difficult to discover;
that/where copies are being made; ***[4 marks max]***

(h) *Award **[1 mark]** for each valid point.*

For example:

- a MIDI controller converts music to MIDI format;
- in real time whilst it is being played;
- attached to the instrument or microphone;
- a controller generates music as a musician is playing; ***[2 marks max]***

(i) *Award **[1 mark]** for each valid point.*

For example:

- a sequencer captures stores and edits music;
- can be a dedicated hardware device or an application;
- the hardware device is wired to carry out these actions;
- application is a set of program instructions;
- the software is slower;
- but can be updated, modified;
- hardware is faster;
- but needs replacing if changes are needed; ***[4 marks max]***

(j) (i) *Award **[1 mark]** for each of the following.*

proximity sensors are used to pick up movement and position;
speed of these two analog measurements converted to pitch and volume;
by MIDI devices;
edited to fit within the norm of music data and output; ***[2 marks max]***

(ii) *Award **[2 marks]** for any reasonable point elaborated and **[1 mark]** for a point stated but not elaborated. There are many possibilities.*

For example:

The music could be very different to what we are used to and hence not have much success ***[2 marks]***.
Could be excitingly different and very successful ***[2 marks]***.
Could revolutionise live performances which would be exactly synchronised and movement could become an integral part of movement ***[2 marks]***. ***[2 marks max]***