International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

MARKSCHEME

November 2010

COMPUTER SCIENCE

Higher Level

Paper 2

13 pages

*This markscheme is **confidential** and for the exclusive use of examiners in this examination session.*

*It is the property of the International Baccalaureate and must **not** be reproduced or distributed to any other person without the authorization of IB Cardiff.*

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 IB Cardiff. 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 that part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

- Each statement worth one 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 markscheme 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. In this subject effective communication is more important than grammatical accuracy.

- 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)   (i)   *Award up to **[2 marks max]**.*
                     Queue is a FIFO data structure;
                     A list in which items may be added only at one end;
                     And removed only at the other end;   *[2 marks]*

         (ii)   *Award up to **[2 marks max]**.*
                     Stack is a LIFO data structure;
                     A list in which one of the ends is designated as the top of the stack;
                     And access (store and retrieve) is restricted to this end of the list;   *[2 marks]*

       (b)   *Example answers:*
                Transfer of data from/to I/O devices;
                Simulation;
                Job queue, order of processing;   *[1 mark]*

       (c)   5;   *[1 mark]*

       (d)   (i)   The queue is empty! ;   *[1 mark]*

         (ii)   *Award marks as follows up to **[4 marks max]**.*
                     *Award **[1 mark]** if the queue is not empty;*
                     *Award **[1 mark]** for temporarily storing the value of the item;*
                     *Award **[1 mark]** if this is at the beginning of the queue;*
                     *Award **[1 mark]** for changing the pointer that points to the end of the
                     queue if the item to be deleted is the last one;*
                     *Award **[1 mark]** for changing the value of the pointer that points to the
                     beginning of the queue;*
                     *Award **[1 mark]** if it points to the next item in the queue;*
                     *Award **[1 mark]** for returning the value that was at the beginning of the
                     queue;*

```
public int dequeue()
{
  if (isEmpty())
  {
    output ("Queue empty");
    return -1;
  }
  else
  {
    int temp=first.item;
    if (first.next== null) // only 1 item in queue
    { last = null;}
    first = first.next // first node removed
    return temp:
  }
}
```

[4 marks]

Question 1 continued

(iii) *Award marks as follows up to [3 marks max].*
Award [1 mark] for deleted item 2.
Award [1 mark] for three items in the queue.
Award [1 mark] for all three correct values (4, 1, 7).

Deleted item: 2
Items in the queue: 4, 1, 7   *[3 marks]*

(e) *Award up to [6 marks max].*
Initialize an empty stack;
While queue is not empty;
   Remove the element from the beginning of the queue/dequeue;
   Push the removed element onto the stack;
While stack is not empty;
   Pop an element off the stack;
   Display it/enqueue it;

Possible answer:
Take the items off the queue;
And put them one by one;
In a new;
Stack;
Now take them off the stack;
And put them back in the queue;   *[6 marks]*

Total: [20 marks]

2.   (a)   (i)   19;   *[1 mark]*

         (ii)   13;   *[1 mark]*

     (b)   *Award [1 mark] for each correct output line, up to [4 marks max].*
           Total number of sales for Salesman 1 is 23;
           Total number of sales for Salesman 2 is 19;
           Total number of sales for Salesman 3 is 29;
           Total number of sales for Salesman 4 is 31;   *[4 marks]*

     (c)   *Award marks as follows up to [6 marks max].*
           Award [1 mark] for correct method heading.
           Award [2 marks] for correct outer loop ([1 mark] for minor error).
           Award [1 mark] for initializing total.
           Award [2 marks] for correct inner loop ([1 mark] for minor error).
           Award [2 marks] for correctly increasing total by Sales[n][m], *([1 mark] for
           an attempt).*
           Award [1 mark] for output.

           Example answer:

```java
public void myst2(int[][] Sales)
{
   for (int m = 0; m < 5; m = m + 1)
   {
      int total = 0;
      for (int n = 0; n < 4; n = n + 1)
      {
         total = total + Sales[n][m];
      }
      output("Total number of sales for Model " + (m + 1) + " is " + total);
   }
}
```
                                                                                          [6 marks]

Question 2 continued

(d)   *Award marks as follows up to **[8 marks max]**.*
*Award **[1 mark]** for correct method heading.*
*Award **[1 mark]** for initializing variables* highestAmount *and* bestSalesman.
*Award **[2 marks]** for correct outer loop (**[1 mark]** for minor error).*
*Award **[1 mark]** for initializing* totalAmount.
*Award **[2 marks]** for correct inner loop (**[1 mark]** for minor error).*
*Award **[2 marks]** for correct calculation of* totalAmount *(**[1 mark]** for minor error).*
*Award **[3 marks]** for if statement, (**[1 mark]** for condition, **[1 mark]** for each assignment statement (×2)).*
*Award **[1 mark]** for each output (×2).*

Example answer:

```java
public void determineBest(int[][] Sales, double[] ModelPrice)
{
  double highestAmount = 0.0;
  int bestSalesman = 0;

  for (int z = 0; z < 4; z++)
  {
    double totalAmount = 0.0;

    for (int k = 0; k < 5; k++)
    {
      totalAmount = totalAmount + Sales[z][k] * ModelPrice[k];
    }

    if (totalAmount > highestAmount)
    {
      highestAmount = totalAmount;
      bestSalesman = z;
    }
  }
  output("The best salesman is Salesman " + (bestSalesman + 1));
  output("The highest total number of sales is " + highestAmount);
}
```
                                                                        [8 marks]

Total: [20 marks]

3.   (a)   (i)   An organized collection;
Of (related) data items;   *[2 marks]*

(ii)   *Award up to [2 marks max].*
The size of the data set is only limited by the amount of secondary memory available;
Data is permanently stored;
Data can be used as often as desired without being re-entered;
Several input sets stored in files can be processed by a computer program;
Output results produced by one program can be stored and used as input in another program;   *[2 marks]*

(b)   (i)   Unique for the record/two records never have the same data/information in this field;
And it is used to search for a particular record;   *[2 marks]*

(ii)   *Award up to [4 marks max].*
Example of unordered file in which each record can be accessed using sequential access to index;
And direct access to data file;

Logical file organization is a file organization in terms of what data items are stored in the file;
And what operations might be performed;

Physical file organization is a file organization in terms how the data is stored on the backing storage device;
And how operations on data can be performed;   *[4 marks]*

(c)   (i)   To improve efficiency of updating algorithm;
Allowing quicker search for matching record;   *[2 marks]*

(ii)   *Award [1 mark] for each correct step identified in the algorithm, up to [8 marks max].*
Read the first record from `StockFile` (and assign it to `stockRec`/a temporary variable);
Read the first record from `UpdateFile` (and assign it to `upRec`/a temporary variable);

While not end of `UpdateFile`;
Compare `itemNo` in `stockRec` with `itemNo` in `upRec`;
If they match then;
    Update the `quantity` in `stockRec`;
    By subtracting the value of `quantity` in `upRec`;
    Write the updated record into `NewStockFile`;
    Read another record from `UpdateFile`;
    Read another record from `StockFile`;
Else Read another record from `StockFile`;

When the end of `UpdateFile` is reached;
Copy all remaining records in `StockFile` into `NewStockFile`;   *[8 marks]*

Total: [20 marks]

4. (a) (i) *Award up to [2 marks max].*
More robust than other two;
Versatile (can be used for other purposes/buying food, drinks from vending machines on credit basis);
More difficult to fraud;
Hold sophisticated programs that could enhance security: adding PIN numbers, finger print *etc.*; *[2 marks]*

(ii) *Award up to [2 marks max].*
Finger print;
Voice print;
Retina scan; *[2 marks]*

(iii) *Award up to [4 marks max].*
Converts analog input data (such as spoken words, signature identification);
Into computer usable code;
By comparing the patterns inputted;
With a set of pre-recorded patterns;
After the computer matches the patterns;
It executes the appropriate command; *[4 marks]*

(iv) *Award [2 marks] for any comparison, up to [4 marks max].*

STAFF	PASSENGER
Higher level of security because they can enter any area.	Lower level, only restricted areas.
Identification of an employee can be verified using smart cards linked to biometric system.	The only form of identification is passport (can be scanned and matched to the central base).
etc.	

[4 marks]

Question 4 continued

(b) (i)   *Award up to **[1 mark max]**.*
        Operating system
        Software;
        That starts up the computer;
        Coordinates the hardware components;
        The application software programs;
        Controls all the programs;
        Manages the use of primary and secondary storage;
        etc.   *[1 mark]*

    (ii) **Multi-user**
        Many users/clients can be supported at the same time;   *[1 mark]*

    (iii) **Multi-tasking**
        OS can execute more than one task at a time;   *[1 mark]*

    (iv) *Award up to **[3 marks max]**.*
        Needs to see position of airplanes;
        Communicate with the persons on the ground;
        Coordinate all departments such as fire department, first aid, if needed;
        Communicate with pilots;
        etc.   *[3 marks]*

    (v) Many users can check-in;
        At different check-in desks (online check-in, self check-in);   *[2 marks]*

(c) (i) **Batch processing**
        Technique whereby data is collected over a period of time;
        Then is input at one time for storage, or for processing and output;   *[2 marks]*

    (ii) **Real-time processing**
        Immediate processing;
        Each input is processed and there is immediate feedback (action can be
        taken right away);   *[2 marks]*

    (iii) *Award up to **[2 marks max]**.*
        Online processing
        Anything in a computer that is online;
        Is linked up to current processing operation;
        And similarly to real-time;
        Immediate processing;
        Each input is processed and there is immediate feedback (action can be
        taken right away);   *[2 marks]*

(d) (i) Real-time;   *[1 mark]*

    (ii) Batch;   *[1 mark]*

Question 4 continued

    (e)   (i)   *Answer should outline any example of airport systems and an explanation (referring to that system).*
    *Award **[2 marks]** for correct (**[1 mark]** for vague example), up to **[4 marks max]**.*
    'Client' refers to a computer (on which there is a software application) that requests actions;
    'Server' means a computer that provides services for the clients that are available via network;   ***[4 marks]***

    (ii)   *Award **[1 mark]** for any device and **[1 mark]** for its function, up to **[2 marks max]**.*
    Example answers:

    Network card;
    Allows a device to be connected to network;

    Transmission media;
    Data is transferred along cables (wireless tech do not require such a device);

    Switch;
    Connects different parts of network, passes data onto appropriate destination;

    Router;
    Directs data traffic from one to another part of a LAN;

    Gateway;
    Connects a LAN to another network which uses different protocols;   ***[2 marks]***

    (iii)   *Award up to **[2 marks max]**.*
    There are many different computer systems at the airport;
    For transmitting data between different devices/different systems;
    A set of rules is needed (types of error checking, the way of indicating begin and end of data, *etc*.);   ***[2 marks]***

    (f)   *Award up to **[4 marks max]**.*
    A linked list would be appropriate because
    The number of data items on the list is not finite;
    It can hold a sequence of data items that are not necessarily held in contiguous data locations;
    Or in order in which they occur (in Air Traffic Control one piece of information could have higher priority than another);
    Each item in the list (node) contains an information field;
    That may consist of several subfields;
    And a link (address) to the next item on the list;
    The last data item on list indicates there are no further items (null pointer);
    External pointer variable points to the first item on the list;
    New data items can be easily inserted at proper place on the list;
    Data already processed by air traffic controller can be easily deleted;   ***[4 marks]***

Total: [40 marks]