International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

# MARKSCHEME

# November 2011

# COMPUTER SCIENCE

# Standard Level

# Paper 2

11 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 SL Paper 2 Markscheme

**Mark Allocation**

Candidates are required to answer ALL questions *[20 marks]* for question 1, *[20 marks]* for question 2 and *[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 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 marks as follows up to [3 marks max].*
*Award [1 mark] for the first three rows;*
*Award [1 mark] for the next two rows;*
*Award [1 mark] for the final row;*

| i | j |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 0 | 2 |
| 1 | 1 |
| 1 | 2 |
| 2 | 2 |

*[3 marks]*

(ii) *Award [1 mark] for each correct matrix.*

First Pass:

```
2    2    6
3    1    5
4    3    8
```

Second Pass:

```
2    2    6
3    1    3
4    5    8
```

Third Pass:

```
2    2    6
3    1    3
4    5    8
```

*[3 marks]*

(b) (i) Swapping a diagonal element with itself does not do anything so the operation is unnecessary on the diagonal elements;    *[1 mark]*

(ii) *Award [1 mark]* for changing the outer loop to
```
(int i = 0; i < N – 1; i = i + 1);
```
*Award [1 mark]* for changing the inner loop to
```
(int j = i + 1; j < N, j = j + 1);
```

alternatively could add a condition, e.g.
```
if !(i==j)  etc.
```
*[2 marks]*

*Question 1 continued*

(c)   *Award marks as follows up to [4 marks max].*
      *Award [1 mark] for correct method declaration;*
      *Award [1 mark] for correct loop;*
      *Award [1 mark] for correct initialization of internal variable (trace in the*
      *example code) + correct calculation;*
      *Award [1 mark] for correct return statement;*

      *Example answer:*

```
public int traceMatrix(int[][] a, int N)
{
  int trace = 1;
  for (int i = 0; i < N; i = i + 1)
    trace = trace * a[i][i];
  return trace;
}
```
                                                                    *[4 marks]*

(d)   *Award marks as follows up to [7 mark max].*
      *Award [1 mark] for correct method declaration;*
      *Award [1 mark] for correct creation of the sums array;*
      *Award [1 mark] for correct outer loop;*
      *Award [1 mark] for correct initiation of each element of the sums array;*
      *Award [1 mark] for correct inner loop;*
      *Award [1 mark] for correct computation of the sum (sum across rows);*
      *Award [1 mark] for correct return statement;*

      *Example answer:*

```
public int[] rowSums(int[][] a, int N)
{
  int[] sums = new int[N];

  for (int i = 0; i < N; i = i + 1)
  {
    sums[i] = 0;
    for (j = 0; j < N; j = j + 1)
      sums[i] = sums[i] + a[i][j];
  }
  return sums;
}
```
                                                                    *[7 marks]*

                                                          *Total: [20 marks]*

**2.**    (a)    *Award marks as follows up to [4 marks max].*
*Award [1 mark] for correct structure of class declaration;*
*Award [1 mark] for selecting a numeric data type for* parcelID *and* parcelWeight*;*
*Award [1 mark] for selecting String data type for* customerAddr *and* deliveryAddr*;*
*Award [1 mark] for selecting Boolean data type for* isDelivered *and* isPaid*;*

*Example answer:*

```
class Parcel
{
  public int parcelID;
  public int parcelWeight;
  public String customerAddr;
  public String deliveryAddr;
  public Boolean isDelivered;
  public Boolean isPaid;

  ...
}
```
                                                                        *[4 marks]*

   (b)    *Award marks as follows up to [4 marks max].*
*Award [1 mark] for identifying each user type, up to [2 marks].*
*For each user type, award [1 mark] for a reasonable description of the data access they require.*
*e.g.*
*The driver;*
*has to have read/write access in order to update the delivery status;*
*The accountant:*
*only needs read-only access to view the payment statuus:*

                                                                        *[4 marks]*

   (c)    *Award [1 mark] for simply identifying a reason to restrict access, and [2 marks] for a broader explanation, e.g.*
*A user would be restricted to his/her account;*
*as other users' details (address ...) are private;*                    *[2 marks]*

   (d)    *Award marks as follows up to [4 marks max].*
*Award [1 mark] for identifying each procedure, up to [2 marks].*
*For each procedure, award [1 mark] for a reasonable description of the procedure*
*e.g.*
*Keeping login details secret;*
*so no unauthorised person can access;*
*Not leaving their workstation unattended ...;*
*Using a log-in procedure with user name + password...*                 *[4 marks]*

   (e)    *Award marks as follows up to [6 marks max].*
*Award [1 mark] for correct method declaration;*
*Award [1 mark] for correctly obtaining the number of parcels;*
*Award [1 mark] for correct loop;*
*Award [1 mark] for keeping track of max value;*
*Award [1 mark] for correct test for parcel delivered but not paid;*
*Award [1 mark] for correct output;*

*Question 2 continued*

*Example answer:*

```
public void findUnpaid(allParcels[])
{
  int numParcels = allParcels.length;

  for (i = 0; i < numParcels; i = i + 1)
  {
    if (allParcels[i].isDelivered && !allParcels[i].isPaid)
      output allParcels.parcelID;
  }
}
```
*[6 marks]*

*Total: [20 marks]*

**3.**  (a)  (i)  *Award [1 mark] for a simple statement, award an additional [1 mark] for a more detailed explanation.*
Running the new and old systems in parallel;                    *[2 marks]*

(ii)  *Award up to [2 marks max] for simply stating a reason, award up to an additional [2 marks] for more complete and detailed explanations.*
To save money;
To avoid rebuilding the machines;
To be able to test software before the building was done;                    *[4 marks]*

(iii)  *Award up to [2 marks max].*
The consequences of system failure;
The time available;
The cost;
Whether all possibilities will be covered;                    *[2 marks]*

(b)  (i)  *Award [1 mark] for identifying a suitable method and [1 mark] for a more complete outline.*

*Example methods:*
Wi-Fi;
Wired network access (the cart must plug in at gates/kiosks/*etc.*);
Some sort of proprietary radio network;                    *[2 marks]*

(ii)  *Award [1 mark] each for up to **two** potential problems, award up to an additional [2 marks] for more complete and detailed explanations.*
Wi-Fi dead spots;
The airport Wi-Fi network is public;
The cart operator might forget to plug in the data connector;
Proprietary radio network ties you to a particular vendor;
Unauthorized users might plug into the data ports or use the Wi-Fi;                    *[4 marks]*

(iii)  *Award [1 mark] each for up to **two** ways to secure the communications. Award up to an additional [2 marks] for more complete and detailed descriptions.*
Encrypt the data;
Use a secure radio network;
Use a private Wi-Fi network;
Validate each cart's transmissions with a challenge/response protocol;                    *[4 marks]*

*Question 3 continued*

(c)  (i)  *Award [1 mark] each for up to **two** features identified in the case study,*
         *up to [2 marks max].*
         The interface is familiar;
         The interface is easy-to-use;
         Controllers can instantly understand the state of the traffic;
         Controllers can communicate without interrupting each other;
         Controllers know they have a new aircraft when they hear the printer;          *[2 marks]*

    (ii)  *Award [1 mark] each for up to **two** reasonable suggested ways, award an*
          *additional [1 mark] each for more complete descriptions of the*
          *suggestions.*
          Use a touch screen to let controllers move shapes around the display;
          Use an audible signal to indicate a new aircraft;
          Allow controllers to drag and drop aircraft information so they drop it
          onto another controller's display;                                          *[4 marks]*

    (iii) *Award up to [6 marks max]. Answer should include discussion of*
          *three distinct areas. Award up to [2 marks max] for the discussion in*
          *each area. Award [1 mark] if an area is simply mentioned, award an*
          *additional [1 mark] for a fuller discussion.*

          Technical advantages/disadvantages of replacing paper strips, such as:
                Integration with other systems;
                Ability to work when computers fail;
                Availability of records for incident investigations;

          Social issues related to changing to computer-based system, such as:
                Controllers like paper strips;
                Training/re-training requirements;

          The cost and likelihood of failures.                                         *[6 marks]*

                                                                            *Total: [30 marks]*