M10/5/COMSC/SP2/ENG/TZ0/XX/M+

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

MARKSCHEME

May 2010

COMPUTER SCIENCE

Standard Level

Paper 2

11 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 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) *Award up to* ***[4 marks max]****.*

An array;
All of the data is of the same type / easy to inspect or process *etc.* values of an array (in a loop);

Of Boolean;
Only two possible answers;

Of String;
Data is text/characters; ***[4 marks]***

(b) *Award marks as follows up to* ***[4 marks max]****.*
Award ***[1 mark]*** *for initializing total.*
Award ***[1 mark]*** *for correct loop.*
Award ***[1 mark]*** *for updating total.*
Award ***[1 mark]*** *for correct return.*

Example:

```
public int yesTotals()
{
  int total = 0;
  for (int x = 0; x < 100; x = x + 1)
  {
    if (answers[x])  // if(answers[x] = "yes") if String chosen in (a)
    {
      total = total + 1;
    }
  }
  return total;
}
```

[4 marks]

continued ...

Question 1 continued

(c) *Award marks as follows up to **[8 marks max]**.*
*Award **[1 mark]** for initializing the arrays with at least 100 positions*
*Award **[1 mark]** for initializing totals to 0 (only totalB is necessary).*
*Award **[1 mark]** for correct loop.*
*Award **[1 mark]** for correct comparison and assignment for negative integers (array `a`).*
*Award **[1 mark]** for correct comparison and assignment for positive even integers (array `b`).*
*Award **[1 mark]** for correct assignment for remaining integers (array `c`).*
*Award **[1 mark]** for correct use of `if`/`else` throughout.*
*Award **[1 mark]** for correctly incrementing all the way through (only `totalB` is necessary).*
*Award **[1 mark]** for correct value returned.*

Example:

```
public int numberSorter(int[] original)
{
  int[] a = new int[100];
  int[] b = new int[100];
  int[] c = new int[100];
  int totalA = 0;
  int totalB = 0;
  int totalC = 0;
  for (int x = 0; x < original.length; x = x + 1)
  {
    if (original[x] < 0)
    {
      a[totalA] = original[x];  // allow a[totalA++] = original[x];
      totalA = totalA + 1;
    }
    else
    {
      if (original[x] % 2 = 0)
      {
        b[totalB] = original[x];
        totalB = totalB + 1;
      }
      else
      {
        c[totalC] = original[x];
        totalC = totalC + 1;
      }
    }
  }
  return totalB;
}
```

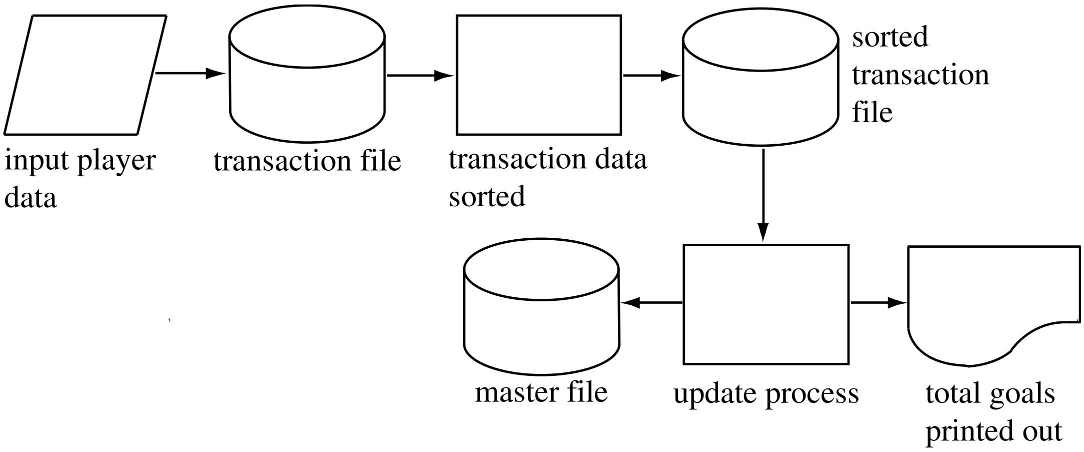
[8 marks]

(d) *Award **[1 mark]** for each point below, up to **[4 marks max]**.*
A new array/object could be created;
With two values;
One for total of array `a` and one for total of array `b`;
This array would be returned;
The data type returned would have to be changed;
The statement that calls this method would have to be changed; ***[4 marks]***

Total: [20 marks]

2. (a) A master file is a **permanent** file whilst a transaction file is **temporary**;
A master file holds the **complete set of data/reference data**, whilst the transaction file contains data that is **temporary / collected over a short period / used to update** the master file; ***[2 marks]***

(b) *Award **[1 mark]** for each correctly labelled symbol up to **[6 marks max]**.
(Award **[3 marks max]** if no arrows are shown).*



[6 marks]

(c) Having the same order allows the update process to be more efficient;
As less comparisons have to be made (or equivalent); ***[2 marks]***

(d) (i) Adams P.;
Do not accept 'Adams'. ***[1 mark]***

(ii) 3; ***[1 mark]***

continued ...

Question 2 continued

(e) *Award marks as follows up to* ***[8 marks max]****.*
Award ***[1 mark]*** *for initialization of variables.*
Award ***[1 mark]*** *for correct outer loop.*
Award ***[1 mark]*** *for correct inner loop that allows for early exit.*
(If both loops are incorrect, award ***[1 mark]*** *if nested loops are attempted).*
Award ***[1 mark]*** *for correct update of goals scored.*
Award ***[1 mark]*** *for correct search (allow even if dot notation is not used). Allow a simple comparison, i.e.* `if (transaction[x].name == master[y].name).`
Award ***[1 mark]*** *for correct update of master file (allow even if dot notation is not used).*
Award ***[1 mark]*** *for correct use of dot notation in two above statements.*
Award ***[1 mark]*** *for efficient incrementation of master file (i.e. not starting at 0 each time).*
Award ***[1 mark]*** *for correct value returned.*

Example:

```
public int update()
{
  int totalGoals = 0;
  boolean found;
  int y = 0;
  for (int x = 0; x < transaction.length; x = x + 1)
  {
    totalGoals = totalGoals + transaction[x].goals;
    found = false;
    while (y < master.length && !found)   // while (!found) is sufficient
    {
      int p = transaction[x].name.compareTo(master[y].name);
      if (p == 0)   // see marking point above
      {
        master[y].goals = master[y].goals + transaction[x].goals
        found = true;
      }
      y = y + 1;
    }
  }
  return totalGoals;
}
```

[8 marks]

Total: [20 marks]

3. (a) (i) Simultaneous operation;
Of two systems (old and new); ***[2 marks]***

(ii) *Award up to* ***[2 marks max]****.*
This allows for the new system to be evaluated;
Without any consequences if the system fails;
As the old system is still functioning; ***[2 marks]***

(b) In order to make the testing as realistic as possible;
To ensure that there would be no problems when the terminal was opened; ***[2 marks]***

(c) *Award* ***[2 marks]*** *for each procedure, up to* ***[4 marks max]****.*
Separate access requirements for different parts of the airport;
This will restrict an illegal access to one section only;

The use of passwords/biometrics *etc.*;
To enter any section;
Note: *Only accept the first biometric example if two are given.*

CCTV cameras;
To allow operators to spot any illegal activity; ***[4 marks]***

(d) (i) *Accept any of the systems that would need real-time.*
e.g. the control tower system; ***[1 mark]***

(ii) *e.g.* for the control tower
Input data must be processed immediately;
In order to have an up-to-date situation;
To prevent any potentially dangerous situations; ***[3 marks]***

continued ...

Question 3 continued

(e) (i) Several people are likely to be in the control tower;
Whose voices could be mistakenly received by the voice recognition system;
Leading to wrong information being input / delays in processing genuine commands / potentially dangerous situations;

Allow answers that focus on the inaccuracy of voice recognition. ***[3 marks]***

(ii) *Award up to* ***[3 marks max]****.*
A touch screen might be suitable;
Commands could be given by touching various menus/boxes;
Different airplanes could be represented graphically on the screen;
Whose position could be changed by touch; ***[3 marks]***

(iii) *Award* ***[1 mark]*** *for a realistic suggestion and* ***[1 mark]*** *for outlining its use.*
Example:
Airplane graphic on screen changes size/colour/makes noise;
To show relative proximity to the airport; ***[2 marks]***

(iv) *Award up to* ***[2 marks max]****.*
The previous system (paper strip) could be kept functioning;
With all data entering both systems;
So that it could take over if needed;

The system could be mirrored;
Using the old control tower / separate server;
With all data entering both systems;

Divert all airplanes;
To another airport;

Do not award any marks for a strategy that would either lose data or would involve a significant interruption in processing. ***[2 marks]***

(f) *There are many possible answers, including ones based on personal experiences. For each side of the argument, award* ***[1 mark]*** *for an indication of the side,* ***[1 mark]*** *for stating why and* ***[1 mark]*** *for an explanation, up to* ***[3 marks max] for each side****.*

Examples:
Governments are ultimately responsible for the safety of its citizens;
Therefore have the right to take any measures necessary;
To identify threats;

This information should not be made widely available;
As it is private/confidential;
And medical details *etc*. could be used against the interests of the passenger;

Governments' proven inability to keep data secure could also lead to a correct answer. ***[6 marks]***

Total: [30 marks]