M13/5/COMSC/SP2/ENG/TZ0/XX/M

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

MARKSCHEME

May 2013

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 penalizing 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 **[2 marks max]**.*
A syntax error could occur (*in Java with mixed mode [int] = [int]*[double])*;
A run-time error could occur in other languages;
Excess memory could be used up (int entered as String);
You would not be able to use the data as intended (int stored as String – cannot use maths or loss of precision); ***[2 marks]***

(b) *eg* a set of yes/no;
responses to a series of questions; ***[2 marks]***

(c) *Award marks as follows up to **[5 marks max]**.*
*Award **[1 mark]** for initialization of total(s);*
*Award **[1 mark]** for loop;*
*Award **[1 mark]** for correct check;*
*Award **[1 mark]** for correct total incremented;*
*Award **[1 mark]** for correct outputs (some text is necessary);*
*Award at most **[1 mark]** for written responses with correct logic;*
Example answer:

```
void howMany()
{
  int yesTotal = 0;
  int noTotal = 0;
  for (int x = 0; x < 100; x++)
  {
    if (array[x] = true)
    {
      yesTotal = yesTotal + 1;
    }
    else
    {
      noTotal = noTotal + 1;
    }
  }
  output (" Yes total =: "+ yesTotal);
  output (" No total =: "+ noTotal);
}
```
[5 marks]

Note: *Solution can just update one total with the other being calculated at the end.*

(d) A 2-D array;
Of real (accept any numerical type); ***[2 marks]***

Question 1 continued

(e) *Award **[1 mark]** per line but do not accept OUTPUT as return*

Example answer:

```
double adams()
{
  double total = 0;
  for (int x = 0; x < 12; x++)
  {
    total = total + array[0][x];
  }
  return total;
}
```

[5 marks]

(f) Store the names in a 1-D array of string;
Each index in this array will link correctly to the same row index in the table; ***[2 marks]***

(g) Change the 4th line to `total = total + array[a][x];`
And pass the variable `a` as a parameter, where it represents the position of the employee in the list;
Accept written answers if the above is correctly outlined. ***[2 marks]***

2. (a) *Award up to **[2 marks max]**.*
The value –1 will not be a valid exam mark;
So it can be used to indicate that that test has not yet been taken;
So as not to make wrong calculations; ***[2 marks]***

(b) *Award up to **[3 marks max]**.*
Calculating the average involves a division;
Which could lead to a decimal value;
Which could lead to point values/numbers;
Decimals are stored as a double; ***[3 marks]***

(c) *Award marks as follows up to **[6 marks max]**.*
*Award **[1 mark]** for initialization of ALL variables;*
*Award **[1 mark]** for correct loop;*
*Award **[1 mark]** for only adds appropriate marks;*
*Award **[1 mark]** for count increased;*
*Award **[1 mark]** for the test;*
*Award **[1 mark]** for average calculated when count `!= 0`;*
*Award **[1 mark]** for correct return;*

Example answer:

```
public double average()
{
  double averageMark = -1;
  int total = 0;
  int count = 0;
  for (int x = 0; x < 15; x++)
  {
    if (marks[x] != -1)
    {
      total = total + marks[x];
      count = count + 1;
    }
  }
  if (count != 0)
  {
    averageMark = total / count;
  }
  return averageMark;
}
```

[6 marks]

Question 2 continued

(d) *Award marks as follows up to **[6 marks max]**.*
*Award **[1 mark]** for correct header;*
*Award **[1 mark]** for return statement;*
*Award **[2 marks]** for correctly calculating average mark for each student (award **[1 mark max]** for a reasonable attempt);*
*Award **[1 mark]** for `temp` created as a `Student` object;*
*Award **[1 mark]** for includes a recognizable bubble sort;*
*Award **[1 mark]** for sort compares average marks;*
*Award **[1 mark]** for correct dot notation used in comparison;*

Example answer:

```
public Student[] bubbleSort(Student[]s)
{
   for (int x=0: x<25; x++)
   {
     s[x].averageMark = s[x].average();
   }
   Student temp;
   boolean swapped = true;
   while (swapped)
   {
     swapped = false;
     for (int i = 1; i < 25; i++)
     {
       if (s[i-1].averageMark < s[i].averageMark)
       {
         temp = s[i-1];
         s[i-1] = s[i];
         s[i] = temp;
         swapped = true;
       }
     }
   }
   return s;
}
```

[6 marks]

Note: *There are several ways of writing a bubble sort.*

(e) *Award up to **[3 marks max]**.*
Note: the example does not encapsulate as private – threat as theory answer
It is safer;
Because using different arrays would make keeping data for one student more difficult to keep together;
Less chance of data being overwritten from other modules;
It is more convenient for programming;
As programming with objects allows advantages of modularity;
eg use of a team of programmers taking specific modules; ***[3 marks]***

3. (a) Its contents change whenever a program using it changes / memory that does have a fixed life;
MUST include example *eg* could contain a downloaded app one moment and then an MP3 file the next; ***[2 marks]***

(b) Start-up instructions;
Accept operating system; do not accept RISC without comment. ***[1 mark]***

(c) With the use of sub-menus;
Selecting one option/icon leads to another set of choices;

AND second example with some comment

By "sliding" the screen sideways;
Revealing more icons/choices; ***[4 marks]***

(d) 3G/4G phones allow users to connect to the Internet;
Users are now downloading files that use a large bandwidth;
Putting a strain on / filling the capacity of the Internet lines; ***[3 marks]***

(e) *Award **[2 marks]** for good discussion of positive examples;*
*Award **[2 marks]** for good discussion of negative examples;*
*Award up to **[2 marks]** for a justified conclusion;*

Examples could include amongst others:

positive:
Quizzes downloaded from the teacher (via Bluetooth)
Real-time research on lesson topic

negative:
Using Internet for cheating
Distraction by social networking sites ***[6 marks]***

(f) The ADC chip is needed to translate digital signals (*eg* music files);
To analogue signals (sound); ***[2 marks]***
*Award **[max 1 mark]** if no example given.*
Allow any example in either direction – must have for full marks.

(g) *Award up to **[4 marks max]**.*
Answer could include the following:
Low power consumption prolongs battery life;
Which is an advantage with mobile devices;
Can only transmit short distances;
Minimizes chance of privacy loss;
Proximity requirements could be inconvenient (give example); ***[4 marks]***

(h) *Award **[1 mark]** for a good example (eg testing water quality).*
*Award up to **[2 marks]** for a clearly explained possible use (eg sensor could be connected to the phone by Bluetooth).* ***[3 marks]***

Question 3 continued

(i) If the phone is lost (*eg* back of taxi);
Then the network provider can disable it preventing data loss/maintaining security;

*Award **[1 mark]** for preventing data loss / help current security issue – ie phone can be disabled if being found to be misused.* ***[2 marks]***

(j) *Award up to **[3 marks max]**.*
A pixel is the smallest element on a screen that can be addressed by the memory;
By increasing the pixel density more elements can be addressed on the same area of screen;
Therefore more data from an image can be displayed in the same area;
Making the image appear sharper;
Although continually increasing resolution will reach a point where the human eye no longer perceives the difference; ***[3 marks]***