N08/5/COMSC/SP2/ENG/TZ0/XX/M+

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

MARKSCHEME

November 2008

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, ***[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.
- The order of points does not have to be as written (unless stated otherwise).
- 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 **[1 mark]** for the correct answer.*
66 ***[1 mark]***

(b) *Award marks as follows, up to **[7 marks max]**.*
*Award **[1 mark]** for correct String parameter and String return.*
*Award **[1 mark]** for declaring and initialising variables.*
*Award **[1 mark]** for test for not found.*
*Award **[1 mark]** for test for end of data marker.*
*Award **[1 mark]** for correct test to find name.*
*Award **[1 mark]** for increment.*
*Award **[1 mark]** for correct return of name.*
*Award **[1 mark]** for return of `0` if not found (need not use an else clause).*

A possible solution is:

```
public String findName(String phone)
{
  boolean found = false;
  int p = 0;
  while( (!found) && (names[p].equals("XXXX")) )
  {
    if (phones[p].equals(phone))
    {
      found = true;
    }
    p = p + 1;
  }
  if (found)
  {
    return (names[p-1]);
  }
  else
  {
    return "XXXX";  // or names[p]
  }
}
```
[7 marks]

(c) *Award up to **[8 marks max]**.*
For the array between the first element and last element;
Compare two adjacent elements;
If the first is larger, then swap;
Move up one element and repeat;
Until `"XXXX"` is reached;
But do not include it in the sort;
Repeat, reducing the length of the array by one;
Until all elements are placed or;
If, on any pass, there are no swaps, terminate the sort; ***[8 marks]***

continued…

Question 1 continued

(d) *Award up to **[2 marks max]**.*
No calculations will be done with the phone number;
There might be a need for additional symbols;
Leading zeroes cannot be represented;
The numbers may be too large to be stored in an int type; ***[2 marks]***

(e) *Award up to **[2 marks max]**.*
An array of record/object/class could be used;
With two fields/data members/(instance) variables;
A 2D array could be used;
With one "column" the names and the other the phone numbers;
As they are both String data types; ***[2 marks]***

Total: [20 marks]

2. (a) (i) *Award **[1 mark]** for each correct definition, up to **[3 marks max]**.*

```
(private) int highTemp;    // accept long or byte
(private) int lowTemp;     // accept long
(private) boolean freezes;
```
[3 marks]

(ii) *Award marks as follows, up to **[4 marks max]**.*
*Award **[1 mark]** for correct method signature (`void` and with an **`int`** parameter).*
*Award **[1 mark]** for correct condition.*
*Award **[1 mark]** for correct assignment to data member of –1 if condition not met.*
*Award **[1 mark]** for correct assignment to class data member.*

A possible solution is:

```
public void setHighTemp(int t)
{
  if ( (t < 5) || (t > 50) )
  { highTemp = -1; }
  else
  { highTemp = t; }
}
```
[4 marks]

(b) *Award marks as follows, up to **[6 marks max]**.*
*Award **[1 mark]** for declaration of loop variables.*
*Award **[1 mark]** for input of city name.*
*Award **[1 mark]** for assignment using array index.*
*Award **[1 mark]** for correct use of the constructor.*
*Award **[1 mark]** for incrementing loop variable.*
*Award **[1 mark]** for looping 5 times.*
*Award **[3 marks]** for a `for` loop with correct syntax.*

Example 1:

```
int c = 0;
do
{
  String name = input("City name: ");
  cities[c] = new City(name);
  c = c + 1;
}while(c < 5);
```

Example 2:

```
static City[] myCities = new City[6];
public static void nameMyCities()
{
  for( int i=0; i < 5; i++)
  {
    myCities[i] = new City(input("Type a City name: "));
  }
}
```
[6 marks]

continued…

Question 2 continued

(c) *Award marks as follows, up to **[7 marks max]**.*
*Award **[1 mark]** for initialising `maxDiff` to `low` value or first value.*
*Award **[1 mark]** for correct initial value for `x` or equivalent.*
*Award **[1 mark]** for correct loop through 20 values.*
*Award **[1 mark]** for calculating new `diff` each time.*
*Award **[1 mark]** for correct `if` statement.*
*Award **[1 mark]** for assigning `diff` to `maxDiff` inside `if` statement.*
*Award **[1 mark]** for output of largest difference.*

A possible solution is:

```
int maxDiff = high[0] - low[0];
for(int x = 1; x < 20; x++)
{
  int diff = high[x] - low[x];
  if (diff > maxDiff)
  {
    maxDiff = diff;
  }
}
output("Max difference: " + maxDiff);
```

[7 marks]

Total: [20 marks]

3. (a) *Award **[1 mark]** for each way (method and appropriate disability) and a further **[1 mark]** for some elaboration up to **[2 marks max]** × 3 = **[6 marks max]**.*
Accept any reasonable but distinct examples.

Screen readers for partially sighted or blind people;
Can be used to read web pages or other text documents;
Can be used to read books/documents with a scanner;

Screen readers for people with touch/dexterity problems;
Can be used to navigate through web pages or other text documents (without having to use keyboard or mouse);

Alert systems for partially sighted or blind people;
Can be used to indicate arrival of email (by reading popup messages);
Can be used to alert them to errors (by reading pop up error messages);
etc. ***[6 marks]***

(b) *Award **[1 mark]** for a step and **[1 mark]** for an elaboration, up to **[2 marks max]** × 2 = **[4 marks max]**.*

Avoid using low contrast or hard to see colours;
Which may affect colour blind users;

Avoid having pages with excessive scrolling;
Which make it difficult for users with dexterity issues;

Provide alt tags for images;
For users using screen readers;

Avoid the use of fancy fonts/very small fonts;
Which can be hard to read for visually impaired;

Avoid the use of sounds with no visual cues;
Which won't be noticed by hearing impaired users;
etc. ***[4 marks]***

(c) (i) *Award **[1 mark]** for an adaptation and **[1 mark]** for an elaboration, up to **[2 marks max]** × 2 = **[4 marks max]**.*

Using keyboards modified to reduce the need for hand movement;
Such as grouping keys together;

Placing more commonly used keys near the centre of the keyboard;
Providing audible feedback on key presses;

Using a replacement for the mouse;
Such as a large trackball which does not require fine hand movement;

Provide cordless, optical mice;
Which are lower maintenance and easier to move;
etc. ***[4 marks]***

continued…

Question 3(c) continued

(ii) *Award **[1 mark]** for an adaptation and **[1 mark]** for an elaboration, up to **[2 marks max]** × 2 = **[4 marks max]**.*

Using keyboards with electronic Braille displays;
So that these users can still "read" the screen;

Speech output system;
To verify the data that is being printed;

Text-to-speech system;
In case printed documents such as schedules and tickets need to be read;
etc. ***[4 marks]***

(d) (i) *Award up to **[4 marks max]**. Accept any reasonable implication.*
Identifying clearly the security method (probably passwords);
They cannot use keyboard input;
They may need to use a head-mouse or similar device;
This usually requires an onscreen keyboard;
Which may make their input of the password obvious to the nurse;

Similar points can be made about, for example, speech recognition systems. ***[4 marks]***

(ii) *Award **[1 mark]** for each problem and a further mark for an elaboration, up to **[2 marks max]** × 2 = **[4 marks max]**.*

Commands may have more than one meaning, *e.g.* "left";
External sources may be confused with the user's voice;
The user might be using a tape-recording, not meaning a command;
Ambient noise may be a problem, drowning out the user's voice commands;
etc. ***[4 marks]***

(iii) *Award up to **[4 marks max]**.*
Voice/sounds are analogue data;
They need to be converted to digital;
By an analogue to digital converter (or by sampling);
Before they can be processed in the computer;
And used to produce suitable output (for the wheelchair); ***[4 marks]***

Total: [30 marks]