



# **MARKSCHEME**

**November 2009**

**COMPUTER SCIENCE**

**Standard Level**

**Paper 2**

*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) Award [1 mark] for each correct column.

highPos	x	x < 8	s[x] < s[highPos]
0	1	true	false
0	2	true	true
2	3	true	false
	4	true	false
	5	true	false
	6	true	false
	7	true	false
	8	false	

[4 marks]

(b) Award [1 mark] for

`if (s[x] > s[highPos])` (this may be expressed in writing)

[1 mark]

(c) Award marks as follows up to [5 marks max].

Award [1 mark] for a correct loop start and finish.

Award [1 mark] for use of `getMinPos` to get the position of the lowest.

Award [1 mark] for a declaration of `temp`.

**EITHER**

Award [1 mark] for a good attempt at swapping array elements.

**OR**

Award [2 mark] for a completely correct swap.

Example:

```
private void sort(int[] s)
{
    for (int x = 0; x < 7; x++)
    {
        int lowPos = getMinPos(s, x);
        int temp = s[lowPos];
        s[lowPos] = s[x];
        s[x] = temp;
    }
}
```

[5 marks]

continued...

*Question 1 continued*

- (d) *Award marks as follows up to [6 marks max].*
  - Award [1 mark] for a return type of `double`.*
  - Award [1 mark] for passing the array as a parameter.*
  - Award [1 mark] for declaring `sum` or equivalent (could be an `int`).*
  - Award [1 mark] for a correct loop through array.*
  - Award [1 mark] for correct summing of elements.*
  - Award [1 mark] for correct calculation of average (do not give for division of `int` by `int`).*
  - Award [1 mark] for return of average.*

*Example:*

```
private double calcAverage(int[] s)
{
    double sum = 0;
    for (int x = 0; x < 8; x++)
    {
        sum = sum + s[x];
    }
    return (sum/8.0);
}
```

**[6 marks]**

- (e) *Award up to [4 marks max].*
  - The array names would also be passed as a parameter;*
  - When a swap is required/done in the scores array;*
  - The names at the same/corresponding locations;*
  - Are also swapped in the names array;*

**Note:** *Accept an algorithm that attempts any of the above marking points (even if it is not completely correct). Each corresponding piece of code is worth [1 mark] up to a maximum of [4 marks].*

**[4 marks]**

**Total: [20 marks]**

2. (a) (i) Award **[1 mark max]** for any of the following: **[1 mark]**  
getName and String/getCd and String/getStars and int;
- (ii) private means that the identifier/method/variable can only be used inside the class/method (where it was declared);  
public means that the identifier/method/variable is available globally/anywhere; **[2 marks]**
- (b) Award marks as follows up to **[4 marks max]**.  
Award **[1 mark]** for correct parameter type.  
Award **[1 mark]** for correct lower limit condition.  
Award **[1 mark]** for correct upper limit condition.  
Award **[1 mark]** for correct logical expression.

*Example 1:*

```
public void setStars(int s)
{
    if ((s < 1) || (s > 5))
    {
        stars = 0;
    }
    else
    {
        stars = s;
    }
}
```

*Example 2:*

```
public void setStars(int s)
{
    stars = 0;
    if ((stars >=0) && (stars <= 5))
        stars = s;
}
```

**Note:** If stars is declared as a local, it will not work (because the Class data member will be masked), but do not penalise for this. e.g.:

```
public void setStars(int s)
{
    int stars = 0;
    if ((stars >=0) && (stars <= 5))
        stars = s;
}
```

**[4 marks]**

*continued...*



Question 2 continued

(c) Range check; **[1 mark]**

(d) *Award up to [4 marks max].*  
**public** Isong( ... ) / the constructor;  
setName;  
setCd;  
setStars;

**[4 marks]**

(e) *Award marks as follows up to [8 marks max].*  
*Award [1 mark] for getting the name of the CD.*  
*Award [1 mark] for getting the name of the CD before the loop.*  
*Award [1 mark] for getting the number of tracks.*  
*Award [1 mark] for using the input number of tracks to size the array.*  
*Award [1 mark] for declaring and initialising x or equivalent.*  
*Award [1 mark] for looping through tracks.*  
*Award [1 mark] for input of correct track data*  
*(both String name and int stars – text prompt detail unimportant)*  
*Award [1 mark] for correctly calling the constructor.*  
*Award [1 mark] for allocating a new instance of class Isong to the array.*  
*Award [1 mark] for incrementing x or equivalent.*

*Example:*

```
String cd = input("What is the name of the album: ");  
int tracks = inputInt("How many tracks will it have? ");  
Isong s[] = new Isong[tracks];  
int x = 0;  
while (x < tracks)  
{  
    String name = input((x+1) + " track name: ");  
    int stars = inputInt("How many stars (1-5)?");  
    s[x] = new Isong(name, cd, stars);  
    x = x + 1;  
}
```

**[8 marks]**

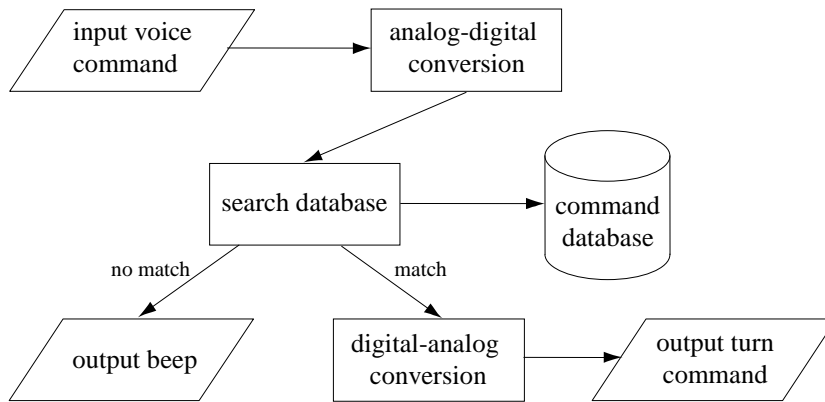
**Total: [20 marks]**

3. (a) (i) Graphical user interface/GUI;  
Command line interface/CLI; [2 marks]
- (ii) Award [1 mark] for a suitable modification and [1 mark] for an elaboration up to [2 marks max] × 2 = [4 marks].  
Examples:  
The GUI could be modified to use an Electronic Braille display;  
As the user will need to move around the screen;
- The CLI could be modified to make use of a speech output system;  
Which can output lines (of text) in sequence;
- Accept any other reasonable modification. [4 marks]
- (b) (i) A macro is a series of recorded commands/keystrokes;  
Which can be played back using a single key; [2 marks]
- (ii) Award up to [3 marks max].  
A number of keystrokes/common operations;  
e.g. typing an address / printing the current page;  
Can be combined under a single/function key;  
Reducing the amount of typing required; [3 marks]
- (c) (i) Award [2 marks] for an advantage and [2 marks] for a disadvantage up to [4 marks max].  
Examples:  
**Disadvantage**  
Sophisticated software may be able to respond to other voices;  
Not only that of the wheelchair owner;  
Possibly causing unintended consequences;
- Advantage**  
More sophisticated software could respond faster;  
This could be an advantage in an emergency;  
And might be able to make use of context;  
Accept any other reasonable advantage/disadvantage [4 marks]
- (ii) Award marks as follows up to [4 marks].  
(Each box must be labelled appropriately.)  
Award [1 mark] for input voice command box.  
Award [1 mark] for conversion boxes (ADC and DAC).  
Award [1 mark] for database search process (including the database itself).  
Award [1 mark] for output turn left command on match.

continued...

Question 3(c) continued

Example:



[4 marks]

(iii) Award up to [3 mark max].

Real time;  
As the wheelchair must respond quickly;  
When an input command is given;  
Otherwise “something bad” may happen;

[3 marks]

(iv) Award up to [3 mark max].

Voice data is analog;  
But the wheelchair (control system) only understands digital data;  
Therefore the voice command must be converted to digital (data);  
The command list is in digital form;  
But the wheelchair (wheels/motors) only respond to analog outputs;  
Therefore the output command must be converted to analog (data);

[3 marks]

(v) Award up to [3 mark max].

The user will not hear any warnings / will not hear the beep;  
Therefore another method of **output** is required;  
e.g. a light/LED;  
e.g. a vibrator/thumper;  
Accept any reasonable suggestion.

[3 marks]

(d) Award up to [2 mark max] for any reasonable points.

Examples:  
A Braille display could be added;  
This may be read with the fingers;  
Audible feedback may be given;  
As fingers touch different parts of the screen;

[2 marks]

Total: [30 marks]