IB DIPLOMA PROGRAMME
PROGRAMME DU DIPLÔME DU BI
PROGRAMA DEL DIPLOMA DEL BI

M06/5/COMSC/SP2/ENG/TZ0/XX/M

# MARKSCHEME

## May 2006

## COMPUTER SCIENCE

## Standard Level

## Paper 2

10 pages

## 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 IBCA. 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 remarking.

**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 three 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, ***[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 part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

- Each marking 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 mark scheme 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. Effective communication is more important than grammatical niceties.
- 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) That the method can be accessed by any other module/program ***[1 mark]*** (not that the method is public).
That no value is returned ***[1 mark]*** (not that it is a void method, or that it is not a function).
***[2 marks]***

(b) Scope refers to the section of the program in which the variable is valid / has a value ***[1 mark]***.
The variable **i** is valid only within the method delete( ) ***[1 mark]***, as it is a local variable ***[1 mark]***. ***[3 marks]***

(c)

| position | i | names[0] | names[1] | names[2] | names[3] | names[4] | names[5] |
|---|---|---|---|---|---|---|---|
| 3 | 4 | sara | surita | juan | nicole | nicole | peter |
| 3 | 5 | sara | surita | juan | nicole | peter | peter |
| 3 | 6 | sara | surita | juan | nicole | peter | peter |

*Award marks as follows*.
***[1 mark]*** *for the position and* ***i*** *columns*.
***[2 marks]*** *for the names array columns,* ***[1 mark]*** *if one mistake is made.* ***[3 marks max]***

(d) The print/display algorithm would only print/display the names up to and including the contents of `names[entries-1]`.
*Award* ***[2 marks]*** *for a correct answer,* ***[1 mark]*** *if the index is incorrect, but the answer is otherwise good.* ***[2 marks]***

(e)
```
public void addEntry(String[]names,String[]numbers,String newName,
                     String newNumber ){
  if (entries !=100) {
        names [entries] = newName;
        numbers [entries]= newNumber;
        entries = entries  + 1;
  }
  else display message "Address Book Full";
}
```

*Award marks as follows*.
***[1 mark]*** *for the if condition*,
***[2 marks]*** *for adding the entries,* ***[1 mark]*** *if the indices are incorrect*,
***[1 mark]*** *for updating the entries variable.* ***[4 marks]***

(f)
```
public String findName(String [ ] names, String [ ] numbers, String
                       callerNumber){
   boolean found = false;
   int i = 0;
   while ( (!found) &&  (i < entries)) {
         if (callerNumber = numbers[i])
               found = true;
         i = i + 1;
   }
   if (found) return names[i-1];
   else return "New Number";
}
```

*Award marks as follows.*
***[1 mark]*** *for indicating in the method signature that a String is returned,*
***[2 marks]*** *for a correct loop,* ***[1 mark]*** *if incomplete but a reasonable attempt,*
***[1 mark]*** *for the comparison,*
***[1 mark]*** *for correctly returning the name if found,*
***[1 mark]*** *for returning a suitable message if not found.* ***[6 marks]***

2. (a) (i) Allows the users to visualize how the final solution might look ***[1 mark]***. They can provide feedback to the design team ***[1 mark]***. ***[2 marks]***

(ii) Gives the design team the chance to explore different possibilities ***[1 mark]***, before choosing the most likely one ***[1 mark]***, or feedback from users ***[1 mark]*** saves wasting time at a later date ***[1 mark]***. ***[2 marks max]***

(b) *Award **[1 mark]** for the sensor and **[1 mark]** for an indication of how it works.*

For example

Weight sensors ***[1 mark]*** could register the passage of a train by its weight triggering the sensor, ***[1 mark]***.
Infra-red sensors ***[1 mark]*** broken as the train passes ***[1 mark]***.
GPS system ***[1 mark]*** uses satellite links to pin-point the position of the train ***[1 mark]***.
***[2 marks max]***

(c) (i) Real-time processing ***[1 mark]*** as signals must be changed immediately ***[1 mark]*** when required. ***[2 marks]***

(ii) Batch processing ***[1 mark]*** as data would be replayed in sequential order ***[1 mark]***.
***[2 marks]***

(d) trainPassed: array of boolean ***[1 mark]***.
signalState: array of integer ***[1 mark]***. ***[2 marks]***

(e)
```
// int totalSignals represents the number of signals on this line.
for (int n = 0; n < totalSignals; n++) {
     if trainPassed[n] {              // true if a train has just passed
       if n > 0
         trainPassed [n - 1] = false; // resets for previous signal
                                      // (unless n = 0)
       signalState [n] = 0;           // sets signal n to red
       if n > 0
         signalState [n-1] = 1;       // sets signal n-1 to
                                      // yellow (unless n = 0)
       if n > 1
         signalState [n -2] = 2;      // sets signal n-2 to green
                                      // (unless n = 0 or n= 1)
     }
}
```

*Award marks as follows.*
***[1 mark]** for checking whether trainPassed is true,*
***[1 mark]** for resetting previous value of trainPassed,*
***[1 mark]** for correctly setting the 3 signals,*
***[1 mark]** for considering at least one of the start of line situations (if n=0, if n=1)* ***[4 marks]***

(f) *Award **[1 mark]** for describing the alternative system, and **[1 mark]** for a further valid comment on the system (e.g. disadvantage/advantage).*

*Award **[1 mark]** for any other system that might be possible but would be unlikely.*
The system could be mirrored with a separate and identical back-up system ***[1 mark]***, although this could prove expensive ***[1 mark]***.
Local control could pass to each stationmaster ***[1 mark]***, although delays would almost certainly result from this ***[1 mark]***. ***[4 marks max]***

3. (a) MIDI stores the data as a series of instructions ***[1 mark]*** whereas CDs and cassette tape files store a digitised version of the actual music ***[1 mark]***. ***[2 marks]***

(b) The MIDI file only requires 2 instructions to be stored (6 bytes) ***[1 mark]***, one which represents the NOTE ON, and the other which represents the NOTE OFF (or NOTE ON with a velocity of 0). However, the CD file would sample and store the sound many times each second ***[1 mark]***. Consequently, the resulting MIDI file is considerably smaller ***[1 mark]***. ***[3 marks]***

(c) They could use the editing function ***[1 mark]*** of the sequencer to correct any errors ***[1 mark]***.
They could record the music at a slower tempo, which would allow them to play the music more accurately ***[1 mark]***, and then play back and re-record the music at the correct (faster) tempo ***[1 mark]***.
The sequencer could be used to add voices (instruments) ***[1 mark]***, which they themselves would be unable to play ***[1 mark]***. ***[4 marks max]***

(d) (i) Channel 4; ***[1 mark]***

(ii) by looking at the MSB (allow the leftmost bit, but <u>not</u> the first bit) ***[1 mark]***, 1 for a status byte and 0 for a data byte ***[1 mark]***. ***[2 marks]***

(iii) 128; ***[1 mark]***

(iv) *Award **[1 mark]** for the first byte and **[1 mark]** for the third byte.*
10011011 00111100 01111111 ***[2 marks]***

(e) (i) Takes up less space / less chance of errors as less characters / easier to read than a sequence of 1's and 0's. ***[1 mark]***

(ii) *Award **[1 mark]** for each correct byte.*
Either 8A 3C 01 using NOTE OFF, or
9A 3C 00 using NOTE OFF with a volume of 0. ***[3 marks max]***

(f) Ring tones ***[1 mark]*** in mobile phones ***[1 mark]*** / sound effects ***[1 mark]*** in computer games ***[1 mark]*** / warning sounds ***[1 mark]*** in burglar alarms ***[1 mark]*** / jingles ***[1 mark]*** in greeting cards ***[1 mark]*** + any other valid answer. ***[4 marks max]***

(g) Multi-timbral ***[1 mark]***: this feature allows a receiving device / sound module / synthesiser to receive and play data from more than 1 channel ***[1 mark]***.
Polyphonic ***[1 mark]***: this feature allows a receiving device / sound module / synthesiser to play more than one note at a time ***[1 mark]***. ***[4 marks max]***

(h) *Award marks as follows up to a maximum of **[3 marks]**.*
sounds such as "oohs" and "aahs" could be handled;
as the notes are reasonably constant;
normal singing, however, contains continuously changing frequencies;
which cannot be captured within the time limitations imposed by the MIDI format;
***[3 marks max]***