



# **MARKSCHEME**

**May 2012**

**COMPUTER SCIENCE**

**Higher Level**

**Paper 2**

14 pages

*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 HL Paper 2 Markscheme

### Mark Allocation

Candidates are required to answer ALL questions [*20 marks*] for question 1, [*20 marks*] for question 2, [*20 marks*] for question 3 and [*40 marks*] for question 4. Maximum total = [*100 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) 25 (km); **[1 mark]**

(b) *Award marks as follows up to [4 marks max].  
Award [1 mark] for correct loop;  
Award [1 mark] for comparison;  
Award [2 marks] for output ([2 marks] if completely correct, [1 mark] if index only outputted);*

*Example answer:*

```
public static circle(int i, int d)
{
    for (int x = 0; x < 30; x = x + 1) // allow x < distances[i].length
    {
        if (distances[i][x] < d) // or distances[x][i]
        {
            output getTownName(x);
        }
    }
}
```

**[4 marks]**

*continued ...*

Question 1 continued

(c) (i) 89 km; [1 mark]

- (ii) Award marks as follows up to [8 marks max].  
 Award [1 mark] for initializing `shortestDistance` to something large;  
 Award [1 mark] for both loops correct;  
 Award [1 mark] for checks `town1` does not equal start town;  
 Award [1 mark] for correct checks in loop 2;  
 Award [1 mark] for correct total;  
 Award [1 mark] for comparison;  
 Award [1 mark] for `shortestDistance` changed if necessary;  
 Award [1 mark] for both towns changed;  
 Award [1 mark] for towns output;

Example answer:

```
public static void threeTowns(int s)
{
    int shortestDistance = 1000;
    int town1 = -1;
    int town2 = -1;
    for (int x = 0; x < 30; x = x + 1) // loop for town 1
    {
        if (x != s) // checks town 1 isn't the starting town
        {
            for (int y = 0; y < 30; y = y + 1) // loop for town 2
            {
                if ((y != s) && (x != y)) // checks town 2 is neither the
                    // starting town nor town 1
                {
                    int total = distances[s][x] + distances[x][y] + distances[y][s];
                    if (total < shortestDistance)
                    {
                        shortestDistance = total;
                        town1 = x;
                        town2 = y;
                    }
                }
            }
        }
    }
    (output shortestDistance;)
    output town1; // allow either indices or town names
    output town2;
}
```

Allow FT if wrong loop condition is used in both (b) and c (ii). [8 marks]

(d) There exists an array of towns (as strings);  
 The distances indices correspond to the indices of the towns array; [2 marks]

continued ...

*Question 1 continued*

(e) *Award up to [4 marks].*

a dynamic data structure / object / array / ArrayList / linked list could be used;  
containing the (correct) town names;  
this structure would then be returned;  
the method signature line would have to be changed to reflect the returning of  
the object;

*[4 marks]*

*Total: [20 marks]*



2. (a) (i)

| English Word | Array Index | Hash Table         |
|--------------|-------------|--------------------|
| antelope     | 0           | antelope, antilope |
| brain        | 1           | brain, cerebro     |
| elephant     | 4           | elephant, elefante |
| fox          | 5           | fox, zorro         |
| door         | 3           | door, puerta       |

[1 mark]

(ii) Because there are only 26 possible positions;  
And there will almost certainly be more than 26 words;

[2 marks]

(b)

| English Word | Array Index | Hash Table (array of linked lists)                    |
|--------------|-------------|---|
| antelope     | 0           | antelope, antilope → null                             |
| camel        | 2           | camel, camelo → null                                  |
| car          | 2           | car, coche → camel, camelo → null                     |
| ant          | 0           | ant, hormiga → antelope, antilope → null              |
| cat          | 2           | cat, gato → car, coche → camel, camelo → null         |
| arm          | 0           | arm, brazo → ant, hormiga → antelope, antilope → null |

[1 mark]

(c) Award marks as follows up to [6 marks].

Award [1 mark] for input of the two words (these can also be passed as parameters);

Award [1 mark] for initialization of new node;

Award [1 mark] for correct use of hash function;

Award [1 mark] for initialization of temp;

Award [1 mark] for dealing with an empty list (reassignment of head);

Award [1 mark] for dealing with a non-empty list (reassignment of head + linking of new node to previous head);

**Note:** the **if ... then ...** clause can be turned around into a single **if ...**

Example answer:

```
public void add()
{
    String englishWord = input("Enter the English word: "); // can enter
    String spanishWord = input("Enter the Spanish word: "); // as param's
    Node newNode = new Node(englishWord, spanishWord);
    int value = hash(englishWord);
    Node temp = listHeads[value];
    if (listHeads[value] == null)
    {
        listHeads[value] = newNode;
    }
    else
    {
        newNode.next = temp; // or newNode.next = listHeads[value];
        listHeads[value] = newNode;
    }
}
```

[6 marks]

continued ...

*Question 2 continued*

- (d) *Award marks as follows up to [5 marks max].  
Award [1 mark] for correct use of hash function;  
Award [1 mark] for initialization of temp;  
Award [1 mark] for correct loop (Java syntax is not needed);  
Award [1 mark] for moving temp on;  
Award [1 mark] for correct output;*

*Example answer:*

```
public void search ()
{
    String english = inputString("Enter the English word: ");
    int value = hash(english);
    Node temp = listHeads[value];
    while (temp.englishWord.compareTo(english) != 0)
    {
        temp = temp.next;
    }
    output("The Spanish equivalent is: " + temp.spanishWord);
}
```

**[5 marks]**

- (e) (i) As the number of words increases, the length of each linked list will also increase;  
Searching the appropriate linked list for the correct (English) word will be sequential;  
And therefore time consuming (which defeats the purpose of having a hash table);
- (ii) A different hash function should be used;  
Which increases the number of hash values (positions) which will reduce the number of collisions and consequently the search time;

**[3 marks]**

**[2 marks]**

*Only award full marks for parts (e)(i) and (ii) if the answers are both clear and show good knowledge. Only allow answers that deal with the actual hashing process.*

**Total: [20 marks]**

3. (a) *Award up to [4 marks max].*  
 It can be searched faster than the main file (in secondary memory);  
 Because it is in the memory (RAM/primary memory) / because the RAM is faster memory;
- Because it is a subset of the main data;  
 It might just contain the data for each record needed for the customer receipt;
- It might point to the correct record in the main file;  
 (Acting as a full index to allow retrieval of data);
- Dynamic so that only the required amount of memory is used;  
 And expand (or contract) as necessary;
- Dynamic because the number of products sold changes over time;  
 A fixed-size data structure might not be big enough; *[4 marks]*
- (b) A binary tree can be searched faster than a linked list / permits binary searching;  
 As each comparison eliminates (potentially) half of the tree / BigO (log n); *[2 marks]*
- (c) (i) The id for the item (being sold) / key field / barcode;  
 A pointer to the address in the main file of the relevant record; *[2 marks]*
- (ii) The item id code;  
 Would be used to search the data structure;  
 Once found the record address pointed to by the found node;  
 Would locate the correct record;  
 The stock level (field);  
 Would then be updated; *[6 marks]*
- (d) As each item is purchased the quantity available is decreased;  
 If this level is less than a preset value;  
 The item is added to a reorder file / an order is automatically made;  
**Or**  
 At the end of each day;  
 The file is searched sequentially to check actual levels against preset levels;  
 If necessary the item is added to a reorder file / an order is automatically made; *[3 marks]*
- (e) *Award up to [3 marks max].*  
 As the items are read into a binary tree;  
 They are compared to the root node;  
 If alphabetically the item is less than the root node the left subtree is inspected, otherwise the right;  
 The process is continued until an empty leaf node is found, and the item is placed there;  
 An in-order traversal will then display all items in alphabetical order;  
**Or**  
 As each item is read into a queue/linked list;  
 It is placed into its correct alphabetical position;  
 When this is completed the data is printed off in order; *[3 marks]*

**Total: [20 marks]**

4. (a) *Award up to [2 marks max].*  
Reduction in size of memory components/processor;  
Reduction in costs of components;  
Use of RISK instruction sets;  
Demand from consumers;  
Reduced power consumption requirements; *[2 marks]*
- (b) Less danger of damage when moving around;  
As flash memory has no moving parts;
- Low energy consumption;  
Prolongs battery life;
- Small (physical) size for size of memory;  
Reduces weight of device / lots of memory for its size;
- Fast processing;  
To permit multi-tasking etc.
- Can be used to supplement permanent memory (as cards);  
Which also allows files to be transferred physically / creates unlimited storage space; *[4 marks]*
- (c) *Award up to [4 marks max].*  
Can efficiently cover a suitably sized area (airport lounge, small building);  
As it does not require line of sight;  
Or cables / it's wireless;  
High bandwidth is available now / high data transfer speed;  
And data travels at the speed of light (between devices);  
Suitable comparison with another technology (*e.g.* Infrared needs line of sight); *[4 marks]*
- (d) *Award up to [4 marks max].*  
Most PCs run Windows operating systems;  
So it is profitable for writers of viruses to write for Windows;  
As different OS require reprogramming virus programs;  
And as there is no such dominance in the smartphone market;  
There are less viruses around for each OS; *[4 marks]*
- (e) (i) Set up each other's phones as "trusted" devices / use of passwords;  
And enable "trusted devices" to communicate between themselves automatically / to allow safe communication between 2 phones;
- Choose the mode that makes the user consciously accept (or decline) any requests;  
Such as file transfer or the opening of files *etc.*;
- Don't accept encryption (students won't do this on a smartphone) *[4 marks]*
- (ii) *Award up to [2 marks max].*  
A music system and its remote/the speakers;  
A pair of customers exchanging files;  
A television with its satellite receiver;  
A computer with a printer;  
Cordless telephone and its receiver;  
*etc.* *[2 marks]*

(iii) *Award marks as follows up to [4 marks max].*

*Award [1 mark] for the idea that different frequencies can be used;*

*Award [1 mark] for other devices could be using the same frequencies;*

*Award [1 mark] for the two on the same piconet switch frequencies;*

*Award [1 mark] for rapidly/often (to prevent/minimize interference);*

*Only award maximum marks for a clear knowledgeable answer.*

A piconet can operate over a small frequency range, however, within this range there are many possible values. When two devices initially make contact they communicate on one of these values. They then (together) change this frequency many times per second to make it unlikely that another piconet will cause any significant interference.

*[4 marks]*

*continued ...*

*Question 4 continued*

- (f) *Award marks as follows up to [6 marks max].*  
*Award [2 marks] for a detailed description of the example/scenario;*  
*Award [2 marks] for sound technical knowledge;*  
*Award [2 marks] for discussing the advantages/disadvantages;*

*One example of many:*

A doctor would be able to access a patient's records from anywhere with Wi-Fi access, for example, from within a different hospital or the patient's house or even the scene of an accident. This will provide the doctor with up-to-date medical information which might be vital.

However, any Wi-Fi network has security issues with the possibility of third party interception of data, especially for unsecured networks where no key is needed. Interception of data could have negative consequences for the patient concerned, as it could lead to problems regarding employment, insurance or even to circumstances in which blackmail may occur if particular conditions become known.

**[6 marks]**

- (g) (i) *Award up to [4 marks max].*  
Could access police database;  
To check details of occupants or car;  
  
Could download photo of suspect;  
To check id;  
  
Could download report;  
To save time with paperwork / time in office;

**[4 marks]**

- (ii) *Many examples including:*  
Fingerprints could be taken by a (second) device equipped with Bluetooth;  
Data is then transferred to smartphone then sent to headquarters;

**[2 marks]**

- (h) *Award up to [4 marks max].*  
Each area of the city is divided into cells;  
A certain subset of the possible frequencies can be used by each cell;  
But adjacent cells cannot use the same frequencies / but users in different cells can;  
Sender and receiver use different frequencies;  
Signal strength is limited so senders (or receivers) in different cells do not get interference;  
If user moves from one cell to another, his/her frequency is changed as appropriate;

*Only award maximum marks if the answer is clear and shows good technical knowledge.*

**[4 marks]**

**Total: [40 marks]**