International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

# MARKSCHEME

# May 2014

# COMPUTER SCIENCE

# Higher Level

# Paper 1

15 pages

**General Marking Instructions**

**1.** Follow the markscheme provided, award only whole marks and mark only in **RED**.

**2.** Make sure that the question you are about to mark is highlighted in the mark panel on the right-hand side of the screen.

**3.** Where a mark is awarded, a tick/check (✓) **must** be placed in the text at the **precise point** where it becomes clear that the candidate deserves the mark. **One tick to be shown for each mark awarded.**

**4.** Sometimes, careful consideration is required to decide whether or not to award a mark. In these cases use Scoris™ annotations to support your decision. You are encouraged to write comments where it helps clarity, especially for re-marking purposes. Use a text box for these additional comments. It should be remembered that the script may be returned to the candidate.

**5.** Personal codes/notations are unacceptable.

**6.** Where an answer to a part question is worth no marks but the candidate has attempted the part question, enter a zero in the mark panel on the right-hand side of the screen. Where an answer to a part question is worth no marks because the candidate has not attempted the part question, enter an "NR" in the mark panel on the right-hand side of the screen.

**7.** Ensure that you have viewed **every** page including any additional sheets. Please ensure that you stamp 'SEEN' on any page that contains no other annotation.

**8.** A mark should not be awarded where there is contradiction within an answer. Make a comment to this effect using a text box or the "CON" stamp.

**Subject Details:           Computer Science HL Paper 1 Markscheme**

**Mark Allocation**

Section A:  Candidates are required to answer **all** questions.  Total 25 marks.
Section B:  Candidates are required to answer **all** questions.  Total 75 marks.
               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**".

**General guidance**

| Issue | Guidance |
|---|---|
| Answering more than the quantity of responses prescribed in the questions | • In the case of an "identify" question read all answers and mark positively up to the maximum marks.  Disregard incorrect answers.<br>• In the case of a "describe" question, which asks for a certain number of facts *eg* "describe two kinds", mark the first two correct answers.  This could include two descriptions, one description and one identification, or two identifications.<br>• In the case of an "explain" question, which asks for a specified number of explanations *eg* "explain two reasons …", mark the first two correct answers.  This could include two full explanations, one explanation, one partial explanation *etc*. |

**SECTION A**                    *Total: [25 marks]*

1.  *Award up to [2 marks max].*
    Roles/activities of the users (*eg* permissions, security, partitions, collaborative work);
    Resources (HW and SW equipment) appropriate for the organization;
    Costs/budget limits;
    Delivery time;
    Compatibility with the old system (data);

    *(Other acceptable answers are possible, from the economic, operational and technical perspective.)*                    *[2 marks]*


2.  *Award up to [2 marks max].*
    Testing prior to product's full release / last stage of testing;
    To see if it works properly / complete functionality / usability;
    Performed by end users (not by designers);                    *[2 marks]*


3.  *Award up to [4 marks max].*
    *Award [1 mark] for the identification of an advantage and [1 mark] for explaining the advantage.*
    *Award [1 mark] for the identification of a disadvantage and [1 mark] for explaining the disadvantage.*

    *Example Advantages:*
    Can highlight aspects that are not detected in questionnaires/interviews;
    So the observer can help produce more detailed reports;

    Observations may be more reliable than interviews;
    Because they can reveal what people actually do instead of what they say they do;

    *Example disadvantages:*
    Time consuming / expense;
    The observer might need to observe a complete cycle *etc* which could last a significant amount of time;

    If the observations are made by only one person, they may be biased;

    Observations may be unreliable;
    Because people act differently when they know they are being watched.
    *(Accept formulations that express similar or plausible ideas.)*                    *[4 marks]*

**4.** *Award [1 mark] for identifying an issue and [1 mark] for an explanation.*
Size of screen;
Therefore difficult to see / use (in poor light);

Size of keys;
Therefore difficult to access functions;

Battery life;
May need to recharge regularly;

Touch screen keys on tablets *etc*;
Lack of tactile feedback; *[2 marks]*

**5.** *Award up to [2 marks max].*
*Award [1 mark] for identifying **two** types of primary memory.*
*Award [1 mark] for the use of **each** type of the memory identified ×2.*

RAM stores data and instructions currently in use
ROM stores permanent instructions
Cache stores frequently used instructions

*(Award [1 mark] if only general scheme of CPU is given.)* *[2 marks]*

**6.** Initially compare with node pointed to by the head;
(If not correct place) move through list using pointers until correct alphabetical position is found;
Adjust pointers accordingly;

*(Drawings are acceptable, but award marks only if they clearly show how pointers are correctly rearranged, following the three guidelines above.)* *[3 marks]*

**7.** A colour will be split into three components *(Accept RGB as an example)*;
Each component will be assigned a certain number of bytes; *[2 marks]*

**8.** *Award up to [2 marks max].*
*Examples of features:*
No central server;
Resources are more widely available (storage, bandwidth, computing power);
Redundancy/recovery;
Supports file sharing for collaborative work; *[2 marks]*

**9.**     *Award up to **[2 marks max]**.*
It is used in the formation of virtual memory / use of secondary memory;
To increase the amount of primary memory;
Memory divided into (tagged) "pages";
Which are then transferred in and out as required;                           *[2 marks]*

**10.**    *Award **[1 mark]** for a feature and **[1 mark]** for a description, for two features, up to **[4 marks max]**.*
*Feature:* Autonomy;
*Description:*
Agents *activate* alone for a task and are not invoked for a task;
Agents can *select the task* themselves (based on priorities or goal-directed search) without human intervention;

*Feature:* Reactive behavior;
*Description:*
Agent senses the environment in which it is, and decides what to do reacting on its perceptions;

*Feature:* Concurrency/sociality;
*Description:*
Agents can interact with other agents through communication, in different modes: coordination, cooperation, competition;

*Feature:* Persistence;
*Description:*
The code describing an agent runs continuously like a process, and is not executed on demand;                           *[4 marks]*

**SECTION B**     *Total: [75 marks]*

**11.** (a) *Award up to [2 marks max].*
Elderly;
Disabled;
Commuters;
*Accept other reasonable answers.*     *[2 marks]*

(b) *Award [1 mark] for advantage and [1 mark] for explanation, for two items, up to [4 marks max].*
Improved convenience;
One can better control times of functioning, hence costs;

Improved comfort;
One can program the functions according to their specific needs;

Energy efficiency;
One can program/plan the functions based on the surrounding environment and reduce energy waste (interconnected systems);

Safety;
Programmed in a way to avoid electric overload and faults;

*Accept other reasonable answers.*     *[4 marks]*

(c) *Award up to [6 marks max].*
*Award up to [3 marks max] for each of the two ways.*
*Award [1 mark] for the way of access and up to [2 marks] for two additional points, which could be advantages or disadvantages.*
*Examples include fixed/non-fixed installations, digital/analogue.*

Computer based/TV-based control;
The user always knows where the control is;
Because the device may not be portable;
Inconvenient if computer/TV already being used by someone else;

Touch screen/keypads in fixed installation;
Ergonomic gadget/small dimensions/cheap device;
Wide range of different designs to fit different locations in a house;
Difficult to lose;
Inconvenient to operate some devices by having to go to a fixed panel/not suitable for extended use to operate TVs etc;

Smartphone;
Portable/useful for some groups of users (limited mobility);
Can be easily extended to include other devices;
Could be lost or misplaced;
Requires internet/Wi-Fi/signal to operate;
*Accept other suitable answers.*     *[6 marks]*

*Question 11 continued*

    (d)   *Award up to [3 marks max].*
         Transmission: integrated wiring or internet or wireless;
         Requires: extended/dedicated network and hw/sw for protocols/transmissions and sensors/actuators;
         Use: The farmer can vary the parameters/environmental conditions from home at any time and better concentrate on other activities (*eg* trade with KFC);   *[3 marks]*

                                                                                   **Total: [15 marks]**

**12.**  (a)   Data is in the cloud/computing infrastructure;
         SW necessary for the activities is in the cloud;
         Access to SW is with thin client (terminal/computers) by web browsing (on the extranet);   *[3 marks]*

    (b)   *Award up to [6 marks max].*
         Security in storage;
              Data is stored in the server of the service provider;
              The organization has no direct control of its data;
              Legislation in the country of the provider may be weaker than in the user's country;
              Cases of provider's corruption/bankruptcy/data loss are a risk to the organization;

         Security in transmission;
              Applications running in-site may require data in SaaS;
              Hence longer transmission times and higher risk of failure/attack/interception;   *[6 marks]*

    (c)   *Award up to [2 marks max].*
         An external extension to a company's local network;
         Limited access;
         Uses internet protocols;   *[2 marks]*

    (d)   VPN authenticates the sender before (establishing the tunnel);
         VPN access is always encrypted, whereas extranet has limited encryption;
         VPN transmission is always encrypted;
         VPN users have access to everything whereas extranet users only have access to (enabled) specific services;   *[4 marks]*

                                                                                   **Total: [15 marks]**

**13.**   (a)   90.2                                                                                               *[1 mark]*

     (b)   Frequencies less than 100 take a 0 on the left (*eg* 88.7 becomes 088.7);
           Convert each digit into a char to get a string;
           *Allow the "dot" to be omitted in the interpretation.   There is always only one*
           *decimal in the example.*                                                              *[2 marks]*

     (c)   *Award  up to **[6 marks max]**.*

           ***Example answer*** *(searches for the min and max, and then the range is calculated)*
           *Award **[1 mark]** for each of the following*
           Initialization;
           Loop;
           Correct if statement (min);
           Correct if statement (max);
           Compute the range;
           Output the range;

```
MIN = Radio[0]
MAX = Radio[0]
K=1
loop while K<=5
     if Radio[K]<MIN then
          MIN=Radio[K]
     else if Radio[K]>MAX then
          MAX=Radio[K]
     endif
     K=K+1
endloop
RANGE=MAX-MIN
output RANGE
```

           ***Example answer*** *(sorts the array Radio, and then the range is calculated, any*
           *sorting algorithm is acceptable)*
           *Award **[1 mark]** for each of the following*
           Idea of nested loops;
           Correct loops;
           Correct comparison;
           Correct exchange;
           Compute the range;
           Output the range;

```
loop for  K=0 to 4
    loop for  J=0 to 4
     if Radio[J]> Radio[J+1]
       then
           swap Radio[J]and Radio[J+1]
      endif
    endloop
endloop
RANGE= Radio[5]- Radio[0]
output RANGE
```

                                                                       *[6 marks]*

*Question 13 continued*

(d)  *Award up to [6 marks max].*
     *Possible solution below, marks to be awarded following the comments in code*
     *(there are 7 marks at least).*

```
// Q is given and initially empty
I=0
loop while I<6                  //across all rows
    ITEM=Stats[I,0]             //retrieve the button
    Q.enqueue(Radio(ITEM))      //enqueue the frequency
    I=I+1
endloop

if input(Flick) then            //if a flicking occurs
    TEMP=Q.dequeue()            //dequeue an element from Q
    output(TEMP)                //to output it and
    Q.enqueue(TEMP)             //circular use of the queue
endif
```
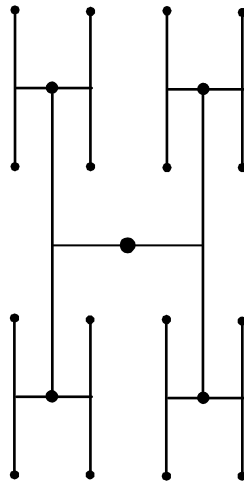                                                            *[6 marks]*

*Total: [15 marks]*

**14.** (a)  *Award up to [3 marks max].*
*Award [1 mark] for a loop (while).*
*Award [1 mark] for correct condition/logical expression.*
*Award [1 mark] for correct operations.*

```
loop while NOT POINTS1.isEmpty()
    CENTRE = POINTS1.pop()
    drawH(CENTRE, SIZE)

end loop                                                    [3 marks]
```

(b)  *Award [1 mark] for correct number and placement of H's;*
*Award [1 mark] for reasonable attempt to show reduced size of successive H's;*



*[2 marks]*

*continued …*

*Question 14 continued*

(c)   *Award up to [6 marks max] as follows:*
      *[1 mark] for the idea that there needs to be a loop that executes once for each generation, **with an additional [1 mark]** if it is implemented correctly, for [2 marks max].*

      *[1 mark] for the idea that the endpoints returned while drawing one generation must be stored for use in drawing the subsequent generation, **with an additional [1 mark]** if correctly implemented, for [2 marks max]. Note: Any valid storage mechanism is permissible ie it does not have to be a stack.*

      *[1 mark] for correctly drawing a generation of H's.*

      *[1 mark] for correctly scaling the size for each generation.*

      *Example answer:*
```
///////////////////
SIZE = 20                               //do not award marks
CENTRE = the middle of the user's display
POINTS1 is a stack, initially empty   //for this part of the
ENDPOINTS = drawH( CENTRE, SIZE )      // of the algorithm
loop COUNT from 0 to 3
     POINTS1.push( ENDPOINTS[COUNT] ) // it is given
end loop
SIZE = SIZE / 2                         // in the question paper
///////////////////

POINTS2 is a stack, initially empty
loop GENERATION from 1 to 3
     loop while NOT POINTS1.isEmpty()
          CENTRE = POINTS1.pop()
          ENDPOINTS = drawH(CENTRE, SIZE)
          loop COUNT from 0 to 3
               POINTS2.push( ENDPOINTS[COUNT] )
          end loop
     end loop
     POINTS1 = POINTS2
     empty the POINTS2 stack
     SIZE = SIZE / 2
end loop
```
                                                                    *[6 marks]*

(d)   *Accept any expression appearing in the following equivalences*

      $4*4*4*4 = 4^4 = 256;$                                         *[1 mark]*

*continued ...*

*Question 14 continued*

    (e)   *Award up to [3 marks max] for:*
        *identifying that the recursive algorithm proceeds downwards (showing on one parameter will suffice);*
        *base case of recursion;*
        *recursive call;*

        The recursive algorithm would use the parameters centre, size, generation counting down (not up);
        When the generation reached is 0 no drawing is done;
        Otherwise draws an H and calls recursively itself on the endpoints of a lower generation;      *[3 marks]*
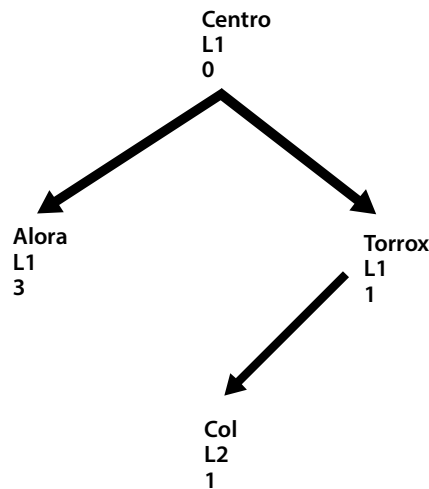
                                                     *Total: [15 marks]*

**15.**   (a)   8;                                            *[1 mark]*

    (b)   *Centro* as root;
        Station names in correct position;
        All 3 items of data for each node;



                                                   *[3 marks]*

    (c)   David;                                           *[1 mark]*

    (d)   5.00 (Euros);
        *Accept 5.*                                   *[1 mark]*

*Question 15 continued*

(e) *Award [1 mark] for each of the following 11 points, up to [9 marks max].*
- *use of nested loops;*
- *use of nested loops with indices that avoid repeating calculations (as shown); (**Note:** outer loop can be to 11 if repeat calculations are avoided, with an IF statement)*
- *correct values retrieved from tree;*
- *check for same line;*
- *check if one of the stations is "Centro";*
- *check and change if negative/ use of absolute value;*
- *correct calculation for same line/one station is "Centro";*
- *correct calculation for different line;*
- *assign value to array;*
- *assign mirror value;*
- *assign value to diagonal;*

```
loop N from 0 to 10
   STATION1 = STATION[N]
   AZ = TREE.getZone[STATION1]
   AL = TREE.getLine[STATION1]
   loop M from N+1 to 11  //start index changed so as not to repeat
                          //code
     STATION2 = STATION[M]
     BZ = TREE.getZone[STATION2]
     BL = TREE.getLine[STATION2]
     if AL = BL or STATION1 = "Centro" or STATION2 = "Centro"
         then               //on same line or passing through "Centro"
            X = AZ – BZ   //number of zones where the travel takes
                          //place can be negative
          if X<0 then     //allow use of absolute
            X = -X        //or equivalent,e.g. X = abs(AZ-BZ)
          endif
          X = X+1
        else  //on different lines
          X = AZ+BZ+1
     endif
     FARES[N][M]=X   //assigns value to 2D array
     FARES[M][N]=X   //assigns mirror value
   endloop
   FARES[N][N]=0       //leading diagonal
endloop
FARES[11][11]=0      //final entry
```
*[9 marks]*

*Total: [15 marks]*