Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

November 2020

Computer science

Higher level

Paper 1

14 pages

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. • 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. • 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

1. (a) *Award **[2 max]**.*
Auto-correct/short sequences/codes;
can be used to represent longer phrases;

Mail merge;
allows production of many letters by only typing the (body of a) letter once;

Voice recognition;
allows easy entry of text;

(b) *Award **[1 max]**.*
Technical documentation explains how to install software;
Technical documentation describes the hardware configuration/operating system needed (to install this software);

2. (a) *Award **[1 max]**.*
B, A, D, C;

(b) *Award **[1 max]***
A, B, C, D;

3. *Award **[3 max]**.*

memory management method that uses secondary memory to increase the amount of primary memory;
transfers data blocks of the same size (“pages”);
from secondary storage to main storage when they are required;
and returns them to secondary storage when they are not;

4. *Award **[4 max]**.*
there is no extra cost;
for running two systems/for extra salaries (no need to increase the number of employees);

benefits can be gained immediately;
because new system is better than the old;

if there is a bug in the new system;
there is not a second system to fall back on/could be disastrous for the company;

employees will need to be capable of using the new system immediately;
without training/with training in advance but not on the system;

Mark as 2 and 2

5. *Award **[2 max]**.*
Reactive behavior;
autonomous agent senses the environment and reacts;

Autonomy;
autonomous agent activates alone for a task / is not invoked for a task / selects the task itself / operates without human supervision;

Persistence;
autonomous agent is a programmed device and the software describing an agent runs continuously;

Sociality;
autonomous agent can interact with other agents through communication / it can work in coordination and cooperation with other agents;

6. (a) *Award **[1 max]**.*
CU/Control Unit;

(b) *Award **[2 max]**.*
MAR (is a register in the CPU that) stores the address of the (next) instruction/data;
to be read from/written to RAM;

7. *Award **[4 max]**.*
*Award **[1]** for looping through array;*
*Award **[1]** for comparing each pair of adjacent items (in the array);*
*Award **[1]** for swapping them if they are in the wrong order;*
*Award **[1]** for repeating this until no swaps is required (which indicates that the array is sorted);*

8. *Award **[4 max]**.*
*Award **[1]** for a truth table with all 8 inputs and up to 3 correct outputs.*
*Award **[2]** for 4 or 5 correct rows in the truth table.*
*Award **[3]** for 6 or 7 correct rows in the truth table.*
*Award **[4]** for all 8 correct rows in the truth table.*

A	B	C	(A XOR B) AND NOT C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Section B

9. (a) *Award **[3 max]**.*
controls the flow of data in the network;
inspects address of data packets;
directs to the appropriate network path/ selects a path between networks (by inspecting address of data packets);
securely transmits data packets (across that path toward the intended destination);

(b) *Award **[4 max]**.*
malicious software/spyware/malware / viruses;
attacking the system;

the type of connection;
valid outline for the slow down (depending on the type of connection);

***Note**: There are many types of internet connection: wireless connections, wireless hotspots, mobile internet, broadband connections, fixed wireless and satellite connection, cable, DSL;*

hardware malfunction;
internet speed can considerably slow down if modem-router configuration is not correct / updated;

external factors;
during the peak hours of daytime the most visited websites encounter more network traffic than what has been anticipated;

***NOTE**: Accept examples. For example, too many students try to connect to an e-learning site at the same time.*

Mark as 2 and 2

(c) *Award **[4 max]**.*
Preventing unauthorized access/No access without user names/passwords;
which should be regularly changed/made difficult to guess;

Data should be encrypted;
Allowing only those with decrypting code access;

Teachers/students should be trained for safe practices;
to create a risk-conscious /security-aware culture within the school;

Installation of virus checkers/spyware software;
to prevent damage to data files or the system / to prevent data being extracted from the files/system;

Mark as 2 and 2

***Note:** Accept appropriate measures such as physical security measures / access control / data security / network and communication security / user awareness and education, etc.*

(d) *Award **[4 max]**.*
*Award **[1]** for looping through the inventory file;*
*Award **[1]** for reading the supply data record (ID number, name, maximum quantity, minimum quantity and remaining quantity);*
*Award **[1]** for checking if the quantity remaining is less than minimum quantity;*
*Award **[1]** If so, calculate the quantity to be ordered (number to buy) / as the maximum quantity minus the quantity remaining;*
*Award **[1]** for outputting the name/ID number of the office supply and the number to buy;*

Example 1:

```
while not end of inventory file
    read supply data record from file
    if remaining quantity < minimum quantity
      then
         numbertobuy =  maximum quantity – remaining quantity
         output ("Reorder ", ID, name, numbertobuy)
    end if
end loop
```

Example 2:

```
for each item in the inventory loop
  with item do
    if remaining quantity < minimum quantity then
      output "Reorder ", ID, name, (maximum quantity – remaining quantity)
    end if
  end do
end loop
```

10. (a) *Award **[2 max]**.*
complexity / simplicity / amount of effort to get a result / number of errors with the time taken to move past them;
readability / comprehensibility / reading or writing speed;
learnability / time to accomplish tasks on the first use;
effectiveness (user performance);
efficiency (time needed to complete a task);

(b) *Award **[2 max]**.*
Award marks for the reason/accessibility issue arising from visual / hearing impairments / cognitive disabilities, and not only for users with disabilities but also for those using mobile devices (alternative browsing devices such as TVs, watches, etc), or those with slow network connections.
Visual display design is not logical;
workflows are not simple, and do not require as few interactions as possible to complete;

Visual display design is not consistent;
navigation, header, footer, and main content are not always in the same places;

Visual display design is not usable as possible;
tools are not easy to use, processes are not broken down into logical steps;

People with poor eye-sight/color blind can have difficulties to distinguish;
because of the way graphics, words and directions are used / combinations of some colours (*eg*, red and green);

(c) (i) *Award **[1 max]**.*
RAM/primary memory;
It's held by the process/thread handling the customer access, which may be in RAM or in other storage if the process is swapped out while handling the access;

(ii) *Award **[4 max]**.*
OS (memory management function) allocates / deallocates memory to each process/customer task, and guarantee each customer task the resources it needs to run correctly;
moves processes back and forth between main memory and secondary memory during execution to prevent overwriting / accidental interchange;
OS (hardware memory protection- part of an OS) isolates/protects customer's data/applications;
and control access rights to the specified memory area (for example, prevents write access to the memory which is not allocated to the process/ customer task);
and protects data / applications when in memory / RAM from malicious code (prevents attempts to execute the contents of the partition/ allocated memory);

(d) *Award **[6 max]**.*
privacy of customer/ person data;
company needs to gain permission from customer;
customer would need to be able to view all data details;
customer must be informed about all uses that will be made of data;
customer must be informed to whom data will be disclosed.
legal issues to do with unauthorized disclosure of customer data;

11. (a) *Award **[4 max]**.*
Proximity sensors/ range sensors;
Which are used to determine how close an object is to the sensor;

Optical sensors /Photocells and other photometric devices (often used in conjunction to proximity sensors);
which are used to detect the presence or absence of objects;

Tactile sensors / Contact sensors / Bumpers;
which are used to determine whether contact is made between sensor and another object;

Touch sensors;
which indicate when contact is made;

Force sensors;
which indicate the magnitude of the force with the object;

Machine vision;
which is used in robotics for inspection / parts identification / guidance (*accept other uses*);

Mark as 2 and 2
Award marks for other miscellaneous category of sensors which also are used in robots. For example, devices for detecting / measuring temperature / fluid pressure / fluid flow / electrical voltage / other physical properties/.

(b) *Award **[3 max]**.*
Output transducer is a device that accepts a (digital) signal from processor;
and turns it into a physical movement;
to make the floor cleaning robot move in different direction;

(c) (i) *Award **[1 max]**.*
Any biometric device (finger print, eye scanner);
pin pads/Key pads on doors;
smartphone access(cloud-based) / mobile access control;

(ii) *Award **[3 max]**.*
The answer to partc (ii) should match the candidate's answer to part (i).For example:
a camera is used to scan the iris / finger print is scanned;
a database stores images (scanned iris/finger print of each employee) and the rooms they are allowed access to;
computer compares the scanned image to images stored in the database;
if found, the employee is allowed to enter;

(d) *Award **[4 max]**.*
Polling requires the computer to actively interact with each swipe device;
the computer periodically checks each swipe device whether it has requested service,
if it does not require servicing, the computer examines the next one, *etc*. If one of them requires servicing, the computer switches to running the serving program;
polling will waste processing time whilst the device is idle;

whilst:
interrupt requires the device to flag the computer;
processor receives interruption signal and services the interrupt by calling the appropriate system subroutine for interrupt processing/interrupt handler;
and hence the computer's time is not wasted whilst the swipe device is idle;

12. (a) *Award **[2 max]**.*
In a stack, the same end is used to insert and delete the elements;
whilst two different ends are used in the queue to insert and delete the elements;

Stack has only one open end so only one pointer is used to refer to the top of the stack;
but queue has two open ends and two pointers should be used (to refer front and the rear end of the queue);

Queue is a first-in-first-out (FIFO) data structure;
and stack is last-in-first-out (LIFO) data structure;

(b) (i) *Award **[1 max]**.*
(`enqueue`) adds an item to rear/tail of the queue;

(ii) *Award **[1 max]**.*
(`dequeue`) removes an item from front/head of the queue;

(iii) *Award **[1 max]***
(`isEmpty`)
checks if the queue is empty or not;
returns `TRUE` if the queue is empty, `FALSE` otherwise;

(c) *Award **[5 max]**.*
*Award **[1]** for a while loop (through queue);*
*Award **[1]** for placing each item removed from the beginning of the queue to the top of the stack;*
*Award **[1]** for a while loop (through stack);*
*Award **[1]** for adding each element popped from the stack to the end of the queue;*
*Award **[1]** for the correct use of stack and queue methods;*

Example:

```
//S is a new/created empty stack
                       // push all of the elements of Q to stack
while not Q.isEmpty()
        X=Q.dequeue()
        S.push(X)          // OR   S.push(Q.dequeue())
end while

                     //then, pop each element from the stack and add it to Q

while not S.isEmpty()
        X=S.pop()
        Q.enqueue(X);      //OR  Q.enqueue(S.pop())
end while
```

(d) *Award **[3 max]**.*

```
mystery(7)= 3 + mystery(4)      ;
          = 3 +3 + mystery(1)    ;
          = 3 +3 + 3 +mystery(-2)=3+3+3+3=12    ;
```

***NOTE**: Award only [**1**] if the working is not shown and only the final result is given.*

(e) *Award **[2 max]**.*
More difficult/complicated to code;
if the data structure being processed is not recursive;

It is difficult to find bugs;
particularly while using global variables;

Can use more memory (than iterative solution) when executed;
because every recursive call increases the call stack;

Recursion to a deeper level will be difficult/impossible to implement;
if the call stack space on the system is limited;

Slower execution / using a recursive function takes more time to execute;
as compared to its non- recursive/ iterative counterpart;

13. (a) *Award **[3 max]**.*
*Award **[1]** for nested loops/loop through all array elements;*
*Award **[1]** for checking the array element for the number of colour;*
*Award **[1]** for updating correctly each of the array elements;*

Example 1:

```
loop for I=0 to N-1
    loop for J=0 to N-1
        if A[I][J]==1
            then A[I][J]=0
            else A[I][J]=1
        end if
    end loop
end loop
```

Example 2:

```
R=0
C=0
I=0
loop while I< N*N
    if A[R][C]==0
        then A[R][C]=1
        else A[R][C]=0
    end if
    C=C+1
    if C>9 then
        C=0
        R=R+1
    end if
    I=I+1
end loop
```

(b) (i) *Award **[1 max]**.*

270 (degrees);

(ii) *Award **[2 max]**.*
A rotation by 360 degrees returns the image/matrix to its original value, so no action need be taken. A rotation by 360+N degrees is equivalent to a rotation by N degrees (this logic repeats, so for 90 degree rotations, there are only 3 transformations needed: by 90, by 2*90 and by 3*90);
making more is unnecessary/inefficient;

Unnecessary calls to the method `rotate()` which would make the algorithm less efficient are avoided;
for example, repeating 10 times and repeating 2 (=10 mod 4) times, both return the array holding the image rotated by 180 degrees;

NOTE: A*ccept any appropriate example, with any value of K which is greater than or equal to 4*;

(c) *Award **[3 max]**.*
*Award **[1]** for completely correct nested loops;*
*Award **[1]** for correctly determining row indexes (in both matrixes, A and B);*
*Award **[1]** for correctly determining column indexes (in both matrixes, A and B);*

***NOTE**: The method heading and return statement may not appear in the answers. Only an algorithm for the method `rotate()` is requested.*

Example:

```
rotate(N,A)
    // initialize two dimensional array B
    // for example, int[][] B = new int[N][N];

      loop for I = 0 to N-1
            loop for  J = 0 to N-1
                 B[I][J] = A[N-J-1][I]
                    // instead of these array subscripts
                    // the following can be written
                    //  B[J][N-1-I] = A[I][J]
            end loop
       end loop

      return B
end rotate
```

(d) *Award **[6 max]**.*

Example:
(Transposes the two-dimensional array and then reverses each row.)

*Award **[3]** for transposing.*
*Award **[1]** for the correct outer loop;*
*Award **[1]** for the correct inner loop;*
*Award **[1]** for the correct swap;*

*Award **[3]** for reversing each row of the transposed matrix.*
*Award **[1]** for the correct outer loop;*
*Award **[1]** for the correct inner loop;*
*Award **[1]** for the correct swap;*
*Award **[1]** for using nothing but a temporary variable to achieve this / no additional / extra space used except one temporary variable;*

***NOTE:** The method heading and return statement may not appear. Only the algorithm for the method `rotate()` is requested.*

```
rotate (N,A)
                        //transposes A
     loop for I=0 to N-1
          loop for J=0 to I-1
             TEMP = A[I][J]
             A[I][J] = A[J][I]
             A[J][I] = TEMP
          end loop
     end loop
                      // reverses each row of transposed matrix A
     loop for I=0 to N-1
          loop for J=0 to N div 2-1
             TEMP = A[I][J]
             A[I][J] = A[J][I]
             A[J][I] = TEMP
          end loop
     end loop
     return A
end rotate
```