Markscheme

November 2016

Computer science

Standard level

Paper 1

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 RM™ Assessor 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 SL 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 45 marks.
            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 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.   *Award up to **[3 max]**.*
Has a small (touch-sensitive) screen;
Uses batteries for power;
No hard disk drive / small memory;
Reliability / Network coverage issues;
Over in warm weather;
Too many steps to access a particular feature;
etc.   **[3]**

2.   (a)   Is used to save time in accessing RAM;   **[1]**

(b)   *Award **[1]** for the cache between RAM and the processor.*
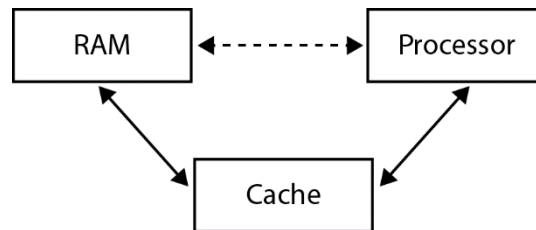
                                                                                                              [1]

3.   *Award **[1]** for stating an advantage and **[1]** for an expansion.*
*Award **[1]** for stating a disadvantage and **[1]** for an expansion.*

Advantages and disadvantages such as:
- Ease of use for mobile users
- Connectivity between different locations
- Reliability
- Cost
- Security
- Change in working patterns
- Health issues
etc.

Example answer:

Advantage:
Ease of use for mobile users;
As they can work in many different locations;

Disadvantage:
Security issues;
As wireless transmissions are easily intercepted;   **[4]**

4.

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

*Award up to **[4 max]** as follows.*
*Award **[1]** for all 8 sets of input values correct.*
*Award **[1]** for correct A XOR B column.*
*Award **[1]** for correct NOT (A XOR B) column.*
*Award **[1]** for correct NOT (A XOR B) AND C column.*   **[4]**

5.   *Award **[1]** for an ethical issue, **[1]** for an explanation, for two issues up to **[4 max]**.*

Points to be discussed:
The data/information is deliberately incorrect;
The data/information has not been validated;
Intellectual property issues;
Plagiarism;   **[4]**

6.   (a)   N;   **[1]**

(b)   *Award **[1]** for each of four correct columns (with headings N, P, N>0 and output).*

X	N	P	N>0	output
2	4	1	true	
	3	2	true	
	2	4	true	
	1	8	true	
	0	16	false	16

[4]

(c)   Calculates X^N;
Note: *DO NOT accept vague answers that may suggest the understanding of N^X or use incorrect terminology*   **[1]**

7.   *Award up to **[2 max]**.*
VPN transmission is always encrypted / provides a secure connection;
Establishes the tunnel between sender/receiver;
Sender/receiver is authenticated before sending/receiving;
VPN users have access to all services available;   **[2]**

Section B

8. (a) *Award [1] for all correct labels or symbols.*
(Accept meaningful, consistent symbols)
Award [1] for correct input flow.
Award [1] for correct output flow.
Award [1] for correct internal processes.
Award [1] for correct dataflow.
Answers given as a process flowchart may still be awarded [1] or [2].
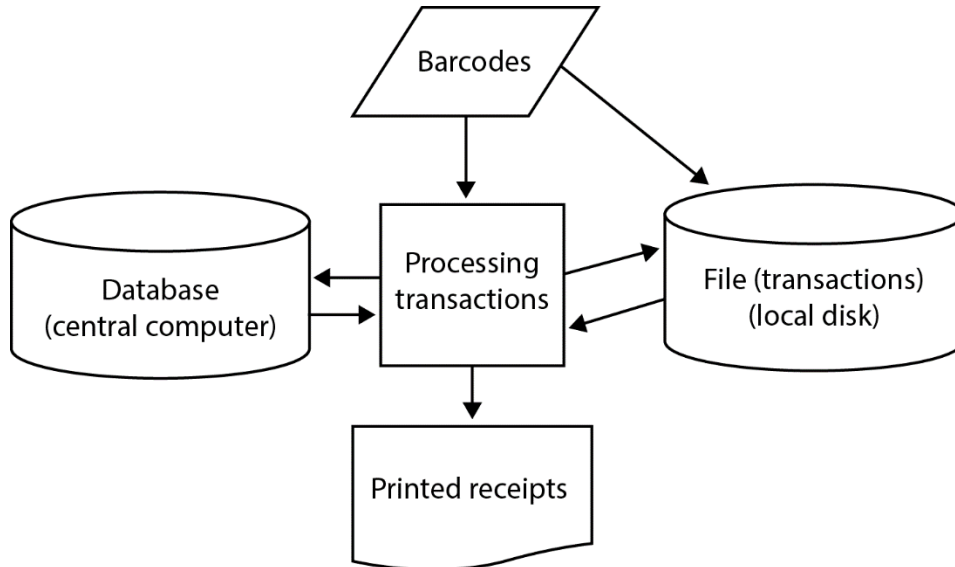
[5]

(b) *Award [1] for identifying a peripheral device, [1] for stating its purpose.*

Example answers:

Keyboard;
To type in some additional data;
Or to type in barcode data when it is not possible to scan;

Magnetic card reader;
Used when a credit card is used;

Microphone;
To call the next customer;
To call manager;

Monitor;
So the salesman can see the information/data on the screen;

Visual display;
So the customer can read the information/data on the display;

Speakers;
For customers to hear information;
For shop assistants to bring another item the customer may wish to buy;   [2]

(c)   *Award up to [2 max].*
        Protocols are sets of rules for transmitting data correctly;
        They ensure that data is sent from a customer's computer and received by the
        shop's computer;
        To create a secure transmission of data from the client to the server through the use
        of the Hypertext Transfer Protocol (HTTPS) *ie* the customer can pay for the books
        securely (using TLS or SSL).   **[2]**

(d)   (i)   Data can be at risk whilst stored on the shop's disk;
                Data can be at risk during transmission;   **[2]**

        (ii)   *Award up to [2 max].*
                All private information must be encrypted;
                Transmission channel must be protected by encryption;
                Logging on to the system must be secured (to prevent intruders);
                Dual data back-up system in case of accidental deletion;   **[2]**

        (iii)   *Award up to [2 max].*
                Details stolen;
                Used for fraudulent purposes;

                Contact details could be shared;
                Used for junk mail/fraud;

                Personal details stolen;
                For identity theft;   **[2]**

9. (a)   *Award up to [2 max].*
                Easy to learn/use;
                Otherwise time may be wasted learning the new language/writing programs in
                this HLL;
                There will be no/less compilation errors;
                There will be no/less logical errors;
                (Reduction of time to create software;)
                Future maintenance/development is possible by other programmers;   **[2]**

        (b)   *Award up to [2 max].*
                GUI;
                Toolbars;
                Menus;
                Built in commands for inputting from touch screens;
                Predicted text so that typing a class name followed by a full stop will bring up a list
                of methods/attributes;
                Automatically use a colour to represent keywords/variables and improve readability   **[2]**

        (c)   *Award [1 max].*
                Help files;
                Online support;   **[1]**

(d) (i) *Award up to [2 max].*
         Must be translated from a higher level language understandable by
         humans/not understood by machines;
         Must be translated into machine code;
         For the CPU to execute it;   **[2]**

     (ii) *Award up to [2 max].*
          Interpreter is faster/immediately warns about syntax errors/executes
          commands and they could use it instead of the compiler while coding and
          debugging their programs;
          Compiler is required when there is a need to produce an executable version of a
          program;   **[2]**

(e) *Award marks as follows:*
    Award [1] for branch of if-then-else leading to correct computation of S=-1;
    Award [1] for the correct loop (boundaries);
    Award [1] for correctly calculating the sum;
    Award [1] for the output;

    Example algorithm 1:
```
if N<=0 then
    S=-1
else
    S=0
    loop for K=1 to N
        S=S+2*K-1
    endloop
end if
output S
```

    Example algorithm 2:
```
if N>0 then
    S=0
    loop for K=1 to 2*N
        if K mod 2==1 then
            S=S+K
        end if
    endloop
else
    S=-1
end if
output S
```
                                                                            [4]

(f) *Award [1] for any of the benefits listed below, [1] for an expansion*
    (ie when/why/who will need it?).

    Reusability;
    Modularity;
    Reliability / All predefined sub-programs are tested and reliable;
    etc.

    Example answer:
    Predefined sub-programs and collections are reusable;
    And this reduces the cost/time needed to develop a large program;   **[2]**

10. (a)   White;   **[1]**

(b)   *Award up to [5 max] as follows.*
Award [1] for input and output.
Award [1] for each correct if statement/colour calculated, up to [3 max].
Award [1] for using modulo arithmetic.

Example answer:
```
X=input('Enter a locker number')
if X MOD 4==1 then
  COLOUR='red'
else
  if X MOD 4==2 then
    COLOUR='white'
  else
    if X MOD 4==3 then
      COLOUR='yellow'
    else
      COLOUR='blue'
    end if
  end if
end if
output(COLOUR)
```

Note: *Accept correct responses that populate an array with 2400 elements, each with the correct colour, and then print the item held in the array. For answers that incorrectly use div rather than mod, allow at MOST one mark FT when returning a 'correct' calculated colour. Do not accept return. Do not accept use of mathematical symbols if not explained*   **[5]**

(c)   (i)   Baber, Ivy;   **[1]**

(ii)   *Award marks as follows, up to [4 max].*
Award [1] for initialization and use of variable (COUNT).
Award [1] for the correct loop.
Award [1] for the correct if statement within the loop
(accept "if PAINTED[K]").
Award [1] for correct COUNT and output.

Example answer:
```
COUNT=0
loop for K=0 to 2399
  if PAINTED[K]== 'true' then
    COUNT=COUNT+1
  end if
end loop
output(COUNT)
```
    [4]

(d)   *Award [1] for stating that the position of a given name should be found in array STUDENT_NAME[].*
Award [1] for using this position to access the elements in the other two arrays.
Award [1] for any named searching algorithm (linear or binary search).
Award [1] for explaining the efficiency – binary search used because STUDENT_NAME[ ] is sorted in alphabetical order.   **[4]**