GCE 2004
June Series

ASSESSMENT and QUALIFICATIONS ALLIANCE

Mark Scheme

Computing
Unit CPT1

Mark schemes are prepared by the Principal Examiner and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation meeting attended by all examiners and is the scheme which was used by them in this examination. The standardisation meeting ensures that the mark scheme covers the candidates' responses to questions and that every examiner understands and applies it in the same correct way. As preparation for the standardisation meeting each examiner analyses a number of candidates' scripts: alternative answers not already covered by the mark scheme are discussed at the meeting and legislated for. If, after this meeting, examiners encounter unusual answers which have not been discussed at the meeting they are required to refer these to the Principal Examiner.

It must be stressed that a mark scheme is a working document, in many cases further developed and expanded on the basis of candidates' reactions to a particular paper. Assumptions about future mark schemes on the basis of one year's document should be avoided; whilst the guiding principles of assessment remain constant, details will change, depending on the content of a particular examination paper.

Further copies of this Mark Scheme are available from:

Publications Department, Aldon House, 39, Heald Grove, Rusholme, Manchester, M14 4NA
Tel: 0161 953 1170

or

download from the AQA website: www.aqa.org.uk

Set and published by the Assessment and Qualifications Alliance.

The Assessment and Qualifications Alliance (AQA) is a company limited by guarantee registered in England and Wales 3644723 and a registered Charity number 1073334. Registered address AQA, Devas Street, Manchester. M15 6EX. *Dr Michael Cresswell Director General*

Computing: Unit CPT1

The following notation is used in the mark scheme

- **; - means a single mark;**
- **/ - means alternative word or sub-phrase;**
- **// - means alternative answer**
- **A – means acceptable creditworthy answer;**
- **R – means reject answer as not creditworthy;**
- **I – means ignore.**

1.	(a)		Operating System// any utility// any user interface// library programs// compiler// assembler// interpreter// translator// driver **A** BIOS **R** any programming language	**1**
	(b)	(i)	software written for a specific/ particular/ certain area/problem/task /function; or by example **R** bespoke software	**1**
		(ii)	appropriate to many areas/tasks/functions/problems/uses; or by example	**1**
	(c)	(i)	designed for the specific task// uses less memory/ disc space// does not have unnecessary features; **R** bespoke software	**1**
		(ii)	able to perform many tasks// can be used for many purposes// cheaper to purchase; **I** Fewer errors	**1**
			Total	**5**

2. (a) (i) machine code; or by description
R Low level language **1**

(ii) assembly code / language; or by description;
R Low level language **1**

(b) Pascal / Delphi/ Visual Basic / Basic / Java / Cobol / Fortran / Algol / or any other 3GL;
R Prolog, Lisp, SQL, HTML, XML
A 'C' **1**

(c) Problem oriented;
less code required// less tedious to program//
one to many mapping to machine code;
portable// machine independent;
Quicker / Easier to understand / write / debug / learn / maintain / modularise;
Data types / structures // structured statements // local variables // Parameters // named variables / constants //
English like keywords / commands / syntax ;
R easy to read
Any two **2**

Total 5

3. (a) (i) Read Only Memory; **1**

(ii) Random Access Memory; **1**

(b) (i) Disk Controller; **1**

(ii) Network Interface Card//Network Adapter;
A Network Card **1**

(c) Address Bus;
Control Bus;
Data Bus; **3**

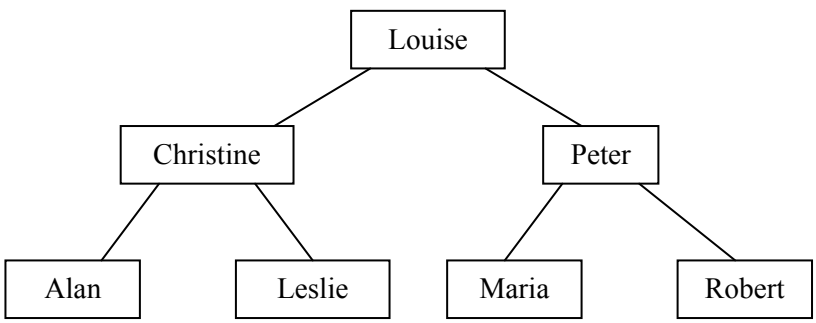
(d) Program stored in main memory;
A RAM/IAS
R ROM
Instructions fetched and executed by processor *(concept)*;
A CPU
Can be replaced by another program;
R cache **2**

(e) (i) The number of 1s (including the parity bit) comes to an even number; **1**

(ii) used to check for errors when data is read / transferred;
Parity bit regenerated / recalculated;
Compared with parity bit;
Any 2 **2**

Total 12

4.



(a) (i) correct position of Louise;
correct position of Christine and Peter;
correct position of Alan and Leslie;
correct position of Maria and Robert;
(If consistent mirror image give full marks) **4**

(b) Root node marked correctly; **1**

(c) Louise, Peter, Maria in correct order (allow follow through from (a)); **1**

Total 6

5.	(a)	(i)	Local Area Network;	**1**
		(ii)	Wide Area Network;	**1**
	(b)	(i)	Intranet // Any example of communication within a building or site; **I** Connection of computers	**1**
		(ii)	Internet// World Wide Web // Any example of communication over a substantial distance; **I** Connection of computers	**1**
	(c)	(i)	A set of rules;	**1**
		(ii)	The rate that signals/ voltage changes are transmitted;	**1**
		(iii)	The number of bits transmitted per second// the number of bits transmitted per time unit; **R** the rate that bits are transmitted; **R** bits of data	**1**
		(iv)	The range of frequencies a medium is capable of transmitting;	**1**
	(d)		Greater bandwidth allows greater bit rate // bit rate proportional to bandwidth;	**1**
	(e)	(i)	Bits transmitted one after the other (along a single channel/ wire/ line); Or by diagram **R** Bits of data	**1**
		(ii)	Bits transmitted (along several wires/ channels/ lines) at the same time; Or by diagram	**1**
		(iii)	Can be transmitted over a longer distance// cabling is cheaper // Less chance of skew;	**1**
		(iv)	Faster transmission;	**1**
				Total 13

6. (a) (i) Analogue to Digital Converter;
A Sound Card **1**

(ii) Stored as a sequence of numeric values;
Each value represents amplitude/height/volume of a signal at that moment;
Sound sampled regularly; **2**

(b) (i) Each pixel represented by a value // image is divided into pixels;
R screen **1**

(ii) Each graphic / drawn element / shape stored individually; **1**

(c) Uses less memory // faster to load / transmit;
Does not deteriorate when enlarged; **2**

(d) (i) Each character stored as a unique code;
Or by example **1**

(ii) ASCII/UNICODE/EBCDIC/BCDIC; **1**

Total 9

7.

Result	Index
0	0
24	1
24	2
57	3
57	4

(a) mark for each correct entry in Result – max 4 marks
A blank as a repeat of the entry above
1 mark for all the entries in Index; **5**

(b) Obtain the largest value; **1**

Total 6

8. (a) (i) Const MaxChars=5; **1**

(ii) Message : Array[1..MaxChars] Of Char//
LastChar : Integer//
Position : Integer//
Found : Boolean;
I var **1**

(iii) Position := Position + 1//
Message[Position] := c//
Position := 0//
Found := FALSE//
Found := TRUE//
Find := Position//
Find := 0; **1**

(iv) If LastChar < MaxChars (Then ….)//
If Message[Position] = c (Then ….)//
If Found (Then …..); **1**

(v) While (Position < LastChar) And Not Found (Do …); **1**

(b) (i) Found;
R var
I type **1**

(ii) Message// LastChar// Position;
R var
I type **1**

(iii) c;
I type **1**

(c) a function always returns one/at least one value //
function name can be used in an expression //
function name can appear on the RHS of an assignment statement //
a procedure can not be used in an expression; **1**

Total 9

END OF CPT1 MARK SCHEME