

# General Certificate of Education 

## Computing 6510

# CPT4 Processing and Programming <br> Techniques 

## Mark Scheme

2008 examination - January series

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 to download from the AQA Website: www.aqa.org.uk

Copyright © 2008 AQA and its licensors. All rights reserved.

## COPYRIGHT

AQA retains the copyright on all its publications. However, registered centres for AQA are permitted to copy material from this booklet for their own internal use, with the following important exception: AQA cannot give permission to centres to photocopy any material that is acknowledged to a third party even for internal use within the centre.

Set and published by the Assessment and Qualifications Alliance.

1 (a) BF2;
1 (b) -;1038;
1
(c) $+; 191 ; .125 ; \quad$ A $1 / 8$

21 mark for sign, 1 mark for value

1 (d) $-; 2 ; .03125 ; \quad$ A $1 / 32$
3 If incorrect part marks as follows
1 mark for complemented mantissa 01000001
1 mark for moving binary point 2 places

1 (e) To maximise precision in a given number of bits;
A to maximise accuracy in a given number of bits
To minimise rounding errors;
To allow a wider range of values to be stored;

2 (a)

| Interactive | Batch | max 1 |
| :--- | :--- | :--- |
| User and computer in two way <br> communication;// User communicates <br> directly with the program while it is <br> running; | Processing continues from <br> beginning to end without user <br> interaction/intervention; | max 1 |
| Program runs with higher priority | Program run in background;// <br> Program runs with lower priority | max 1 |
| Processing carried out as users enter <br> the data; <br> Results are available immediately;Processing delayed until all data <br> have been entered; <br> Results are available when the job <br> is completed; |  |  |

2 (b) Job ID; priority; user name; job delimiters; job completion time; estimated running/processor time; max length of time job can run for; start time of job; main memory required; file size; devices/hardware required (e.g. printer); compiler/assembler/software required; data/source code file required; output destination; what to do on non-successful completion of job; I/O Bound or Processor bound type of batch job;

2 (c) Process requires service from a resource;
Process is timed out// Time slice expires;
Process is pre-empted;

2 (d) Priority queue;
(a) Surface number;

Track/Cylinder Number; Sector/Block Number;

## A layer/platter

A segment/ cluster

3 (b) (i) Memory used for temporary storage of one or more disk blocks; in transit between disk and main memory;

3 (b) (ii) To allow for different speed of devices. Logical records \& physical records of different size;

3
(c) (i) Transfer Completed;

Transfer aborted/failed/timed out;
Handshaking
(c) (ii) Interrupting device/ source supplies; an offset/vector; A index/indexed address added to the base address; A base register

Gives the start address of interrupt service routine/ ISR//
Address vector table cell contains start address of ISR/
R Interrupting device supplies start address of ISR
(c) (iii) a different routine can be easily introduced// routine can be relocated/ dynamically loaded; or words to this effect A The interrupting device only needs to supply a new offset

4 (a)


1 mark for correct boxes
1 mark for correct lines
1 mark for correct line endings
(b)

```
Loan = class
        Public
                                Procedure CreateLoan
                Procedure DeleteLoan
                Procedure GetLoanDetails;
            Private
                Person: Borrower
                BookLoaned: BookCopy;
                DateOfLoan: Time/Date A string
                ReturnDate: Time/Date; A string
            End;
```

1 mark for Loan=Class + Public + Private + End
1 mark for CreateLoan + DeleteLoan + GetLoanDetails
1 mark for Person + BookLoaned
1 mark for DateOfLoan + ReturnDate
A any reasonable names for operations and data items.
(c) Add a new data item ShortLoan; of type Boolean;

A loanlength; integer;
A loantype; string;
Modify the code for the operations;
(a) (i) Each accumulator bit is compared with its corresponding operand bit, if both are 1 the result for this bit position is 1 , otherwise 0 ;
A by example
(a) (ii) AND \#0F; A AND \#CF

Allow ft to (b)

5 (b)

| Label | Opcode | Operand | Comment |
| :--- | :--- | :--- | :--- |
|  | LD | 015 A | Load first character |
|  | AND | $\# 0 \mathrm{~F} ;$ | And convert to a value |
|  | MUL AND \#CF |  |  |
|  | ST | \#10; | Move to upper nibble |
|  | LD | $0155 ;$ | Store in work area |
|  | AND | \#01A6 | Load second character |
|  | ADD | $01 \mathrm{~A} 5 ;$ | And convert to a value |
|  | ST AND \#CF |  |  |
|  | $01 \mathrm{~A} 6 ;$ | Store result |  |
|  |  |  |  |

Or

|  | LD | 015 B | Load second character |
| :--- | :--- | :--- | :--- |
|  | AND | $\# 0 \mathrm{~F} ;$ | And convert to a value |
|  | ST AND \#CF |  |  |
|  | LD | $;$ |  |
| A 01A6 |  |  |  |
|  | AND |  | Store in work area |
|  | MUL |  | And \#CF |
|  | ADD | $;$ | Move to upper nibble |
|  | ST | $;$ | Combine two values |
|  |  |  | Store result |

6 (a) a procedure/routine that calls itself/ is defined in terms of itself;
A Function instead of procedure
$\mathbf{R}$ re-entrant $\quad \mathbf{R}$ program $\mathbf{R}$ iteration

6 (b) (i)

| Procedure Call | T |
| :---: | :---: |
| $\mathrm{P}_{1}$ |  |
| $\mathrm{P}_{2}$ | 18 1 mark |
| $\mathrm{P}_{1}$ |  |
| $\mathrm{P}_{3}$ |  |
| $\mathrm{P}_{4}$ | 11 1 mark |
| $\mathrm{P}_{3}$ | $\left./_{5}^{8}\right\rangle_{11}$ |
|  | $5 \quad 1$ mark |
| $\mathrm{P}_{3}$ |  |
|  |  |



6 (b) (ii) Reverse Inorder// Reverse order; (tree) traversal;
(a) student(jim);
parent(rachel,jim);
male(jim)
female(rachel);
I order
Penalise case once only
(b) mother(X,Y) IF parent(X,Y) AND female(X)
mark for IF and AND
mark for parent (X,Y) and female(X)
Penalise case once only
(c) grandfather(X,Y) IF father(X,Z) AND parent(Z,Y)// grandfather(X,Y) IF male(X) AND parent(X,Z) AND parent(Z,Y) 1 mark for IF and AND
1 mark for father(X,Z) or male(X) AND parent(X,Z)
1 mark for parent(Z,Y)
Penalise case once only

