

### **General Certificate of Education**

## **Computing 6510**

CPT4 Processing and Programming 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
1	(b)	-;1038;		2 1 mark for sign, 1 mark for value
1	(c)	+;191;.125; A 1/8		3
1	(d)	-;2;.03125; <b>A</b> 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 <u>in a given numb</u> A to maximise accuracy <u>in a given num</u> To minimise rounding errors;	<u>er of bi</u> ber of l	<u>ts;</u> bits

To allow a wider range of values to be stored;

#### Max 2

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

Max 2

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	i job; Max 3	1	
2	(c)	Process requires service from a resource; Process is timed out// Time slice expires; Process is pre-empted;			
			Max 2	,	
2	(d)	Priority <u>queue;</u>	1		
3	(a)	Surface number; A layer/platter Track/Cylinder Number;			
		Sector/Block Number; A segment/ cli	ister 3		
3	(b)	(i) <u>Memory</u> used for temporary storage in transit between disk and main me	of one or more disk blocks; nory; 2		
3	(b)	<ul> <li>(ii) To allow for different speed of device Logical records &amp; physical records of</li> </ul>	es. of different size; Max 1	Ĺ	

3	(c)	(i)	Transfer ( Transfer a Handshak	Complete aborted/fa ting	d; uiled/time	ed out;				2
3	(c)	(ii)	Interrupt an offset/ added to	ting devic /vector; 4 the <u>base</u> a	ee/ source <b>A</b> index/i address; A	e supplies; ndexed addres A <u>base</u> register	S			Max 2
			Gives the Address <b>R</b> Interrup	e <u>start add</u> vector tab oting devi	<u>lress</u> of in ole cell co ice suppli	nterrupt servic ontains <u>start ad</u> ies <u>start addres</u>	e routine/ IS I <u>dress</u> of ISR I <u>s</u> of ISR	R//		3
3	(c)	(iii)	a differen routine ca A The in	nt <u>routine</u> an be relo terrupting	can be exponential can be expone	asily introduce ynamically loa only needs to s	ed// ded; <i>or worc</i> upply a new	<i>ls to th</i> offset	his effect	1
4	(a)	Bo	rower	1	1*	Loan	1*	1	BookCopy	
		1 marl 1 marl 1 marl	c for correc c for correc c for correc	et boxes et lines et line end	dings		ı			3
4	(b)		Loan	= clas Publ Pr Pr Priv Pe E E End;	s ic rocedur rocedur rate erson: BookLoa DateOfI eturnD	re CreateLo re DeleteLo Borrower uned: Book Loan: Time Date: Time	Dan Details; Copy; /Date /Date;	A A	string string	
		1 ma 1 ma 1 ma 1 ma	urk for Loa urk for Crea urk for Pers urk for Date	n=Class - ateLoan - son + Boo eOfLoan	+ Public + DeleteL okLoanec + Return	+ Private + En Loan + GetLoa I Date	d nDetails			
1	(c)	A an $\wedge$ dd a	y reasonat	tem Shor	tor operation the second se	ations and data	i items.	an ath:	integer.	4
-	(0)	Modif	v the code	for the or	perations		A loanty	ype; st	ring;	Max 2
5	(a)	(i)	Each acc 1 the resu A by exa	umulator ult for thi mple	bit is con s bit posi	mpared with it tion is 1, other	s correspond wise 0;	ing op	perand bit, if both are	1
5	(a)	(ii)	AND #01 Allow ft	F; AA to (b)	AND #CF					1

6

1

Label	Opcode	Operand	Comment	
	LD	015A	Load first character	
	AND	#0F;	And convert to a value	A AND #CF
	MUL	#10;	Move to upper nibble	_
	ST	01A5;	Store in work area	<b>A</b> 01A6
	LD	015B	Load second character	
	AND	#0F;	And convert to a value	A AND #CF
	ADD	01A5;	Combine two values	
	ST	01A6;	Store result	

Or

LD	015B	Load second character	
AND	#0F;	And convert to a value	A AND #CF
ST	;	Store in work area	A 01A6
LD		Load first character	
AND		And convert to a value	A AND #CF
MUL		Move to upper nibble	
 ADD	;	Combine two values	
ST	;	Store result	

6 (a)

a procedure/routine that calls itself/ is defined in terms of itself; A Function instead of procedure R re-entrant R program R iteration

### 6 (b) (i)

Procedure Call	Т	Output
P <sub>1</sub>	5 $8 $ $18 $ $18$	
P <sub>2</sub>	18 1 mark	18;
P <sub>1</sub>	5 × <sup>8</sup> × <sup>14</sup> × 18	14
P <sub>3</sub>	5 $1$ $1$ mark	
P <sub>4</sub>	11 1 mark	1 mark correct order
P <sub>3</sub>	5 11	8
	5 1 mark	5
P <sub>3</sub>	5 × 8 × 11	
	5 × <sup>8</sup> × <sup>14</sup> 18	

6

6	(b)	(ii)	Reverse Inorder// Reverse order; (tree) traversal;	2
7	(a)	studer parent male(j female I o Penali	nt(jim); t(rachel,jim); jim) e(rachel); order ise case once only	3
		i enun		5
7	(b)	mother mark f mark f	r(X,Y) IF parent(X,Y) AND female(X) For IF and AND For parent(X,Y) and female(X)	2
		I Chang	se case once only	2
7	(c)	grand grand 1 mar 1 mar 1 mar	father(X,Y) IF father(X,Z) AND parent(Z,Y)// father(X,Y) IF male(X) AND parent(X,Z) AND parent(Z,Y) k for IF and AND k for father(X,Z) or male(X) AND parent(X,Z) k for parent(Z,Y)	
		Penali	ise case once only	3