OCR
RECOGNISING ACHIEVEMENT

GCE

Computing

Advanced GCE

Unit **F453:** Advanced Computing Theory

Mark Scheme for June 2011

Oxford Cambridge and RSA Examinations

OCR (Oxford Cambridge and RSA) is a leading UK awarding body, providing a wide range of qualifications to meet the needs of pupils of all ages and abilities. OCR qualifications include AS/A Levels, Diplomas, GCSEs, OCR Nationals, Functional Skills, Key Skills, Entry Level qualifications, NVQs and vocational qualifications in areas such as IT, business, languages, teaching/training, administration and secretarial skills.

It is also responsible for developing new specifications to meet national requirements and the needs of students and teachers. OCR is a not-for-profit organisation; any surplus made is invested back into the establishment to help towards the development of qualifications and support which keep pace with the changing needs of today's society.

This mark scheme is published as an aid to teachers and students, to indicate the requirements of the examination. It shows the basis on which marks were awarded by Examiners. It does not indicate the details of the discussions which took place at an Examiners' meeting before marking commenced.

All Examiners are instructed that alternative correct answers and unexpected approaches in candidates' scripts must be given marks that fairly reflect the relevant knowledge and skills demonstrated.

Mark schemes should be read in conjunction with the published question papers and the Report on the Examination.

OCR will not enter into any discussion or correspondence in connection with this mark scheme.

© OCR 2011

Any enquiries about publications should be addressed to:

OCR Publications
PO Box 5050
Annesley
NOTTINGHAM
NG15 0DL

Telephone: 0870 770 6622
Facsimile: 01223 552610
E-mail: publications@ocr.org.uk

1 mark per bullet point unless indicated otherwise.

Question			Expected Answer	Mark	Additional Guidance
			One mark per bullet point unless indicated otherwise.		
1	(a)	(i)	• while the operating system is loading • when the computer is switched on • after POST	[max 1]	
		(ii)	• provides personal settings	[1]	
	(b)		• use of backing store… • …as additional memory • uses paging / swapping pages (between memory & backing store) • holds part of the program not currently in use • allows large programs to run (when memory size is insufficient)	[max 4]	Allow "uses segmentation" as alternative for bullet 3
	(c)		• a map of where files are stored… • …in backing store/hard disk • provides addresses/pointers to (start of) files • stores file names • stores file sizes • stores access rights • identifies free space • is updated by the operating system when files are saved/deleted • Is used by the operating system when files are accessed	[max 6]	
2	(a)	(i)	• translator	[1]	Accept compiler / assembler
		(ii)	• the original code/code written by the programmer… • …often in a high level language • may be in assembly language • source code can be understood by people… • …but cannot be executed (until translated)	[max 4]	

<table>
<tr><th colspan="3">Question</th><th>Expected Answer</th><th>Mark</th><th>Additional Guidance</th></tr>
<tr><td></td><td>(b)</td><td></td><td>
<table>
<tr><td></td><td>Lexical analysis</td><td>Syntax analysis</td><td>Code generation</td><td>Not during compilation</td></tr>
<tr><td>Optimisation occurs</td><td></td><td></td><td>✓</td><td></td></tr>
<tr><td>Logical errors are detected</td><td></td><td></td><td></td><td>✓</td></tr>
<tr><td>Tokens are created</td><td>✓</td><td></td><td></td><td></td></tr>
<tr><td>Spaces are removed</td><td>✓</td><td></td><td></td><td></td></tr>
<tr><td>Comments are removed</td><td>✓</td><td></td><td></td><td></td></tr>
<tr><td>Incorrect punctuation is detected</td><td></td><td>✓</td><td></td><td></td></tr>
</table>
[1 mark per correct row, max 6]
</td><td></td><td></td></tr>
<tr><td></td><td>(c)</td><td></td><td>Mark band 6-8. High level response.

Candidate has explained in detail why library routines help programmers and has described how they are used.
Information is presented in a clear and organised way.
Candidate has used appropriate technical terminology throughout.
There are few, if any, spelling errors or grammatical errors.

Mark band 3-5. Medium level response.

Candidate has explained why library routines help programmers and has described how they are used, but the response may lack detail.
Candidate has used some technical terminology in the response.
There may be spelling errors or grammatical errors, but they are not obtrusive.

Mark band 0-2. Low level response.

Candidate has listed some relevant points but failed to explain terms or make connections.</td><td>[max 8]</td><td></td></tr>
</table>

Question			Expected Answer	Mark	Additional Guidance
			There is a lack of cohesion in the response. Candidate has failed to use correct technical terms in the response. Spelling and grammatical errors affect the readability of the response. *Points to be made may include:* *Library routines:* • routines are pieces of software… • …which perform common tasks… • …such as sorting/searching • routines are compiled *Why library routines help programmers:* • routines are error-free/have already been tested • already available/ready to use/saves work/saves time • routines may be used multiple times • routines may have been written in a different source language • allows programmer to use others' expertise *How routines are used:* • linker is used… • …to link routine with program • loader handles addresses… • …when program is to be run		
3	**(a)**		• PC holds address of next instruction • PC passes this address to MAR… • MAR holds address of instruction/data • Instruction/data from address in MAR is loaded to MDR • PC is incremented (in each cycle) • PC is changed when there is a jump instruction… • …by taking address from instruction in CIR	**[max 5]**	

Question			Expected Answer	Mark	Additional Guidance
	(b)	(i)	• uses (complex) instructions each of which may take multiple cycles • single register set • instructions have variable format • many instructions are available • many addressing modes are available	[max 3]	
		(ii)	• programs run more slowly… • …due to the more complicated instructions/circuit	[max 2]	
4	(a)	(i)	• exponent 0010 = 2 • mantissa 0.11, move point 2 places to right = 011. * • denary value is (2 + 1) = +3 [*accept any valid conversion method*, max 3]	[max 3]	* FT if exponent incorrect
		(ii)	• exponent 0001 = 1 • mantissa 1.001, move point 1 place to right = 10.01 * • denary value is -2 + ¼ = -1¾ (or -1.75) [*accept any valid conversion method*, max 3]	[max 3]	* FT if exponent incorrect
	(b)		• +5¼ = 101.01 in pure binary • mantissa is 010101, but only 4 bits allowed… • …so mantissa would be 0101 (5 stored, not 5¼) • value stored is inaccurate/precision lost • exponent is 0011 (3) • number would be stored as 01010011	[max 4]	

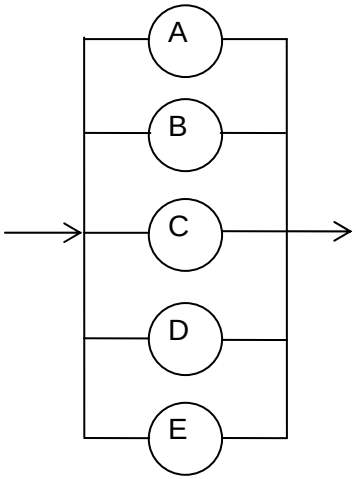
<table>
<tr><th colspan="3">Question</th><th>Expected Answer</th><th>Mark</th><th>Additional Guidance</th></tr>
<tr><td>5</td><td>(a)</td><td></td><td>
<table>
<tr><td>Original set</td><td><u>17</u></td><td>2</td><td>3</td><td>26</td><td>5</td></tr>
<tr><td>Insert 17</td><td>17</td><td><u>2</u></td><td>3</td><td>26</td><td>5</td></tr>
<tr><td>insert 2</td><td>2</td><td>17</td><td><u>3</u></td><td>26</td><td>5</td></tr>
<tr><td>insert 3</td><td>2</td><td>3</td><td>17</td><td><u>26</u></td><td>5</td></tr>
<tr><td>insert 26/no change</td><td>2</td><td>3</td><td>17</td><td>26</td><td><u>5</u></td></tr>
<tr><td>insert 5</td><td>2</td><td>3</td><td>5</td><td>17</td><td>26</td></tr>
</table>
marks for:
• list of sorted numbers is built up…
• …with one number at a time being inserted into correct position
• plus 1 mark per correct row [max 4 rows] *
</td><td>[max 5]</td><td>* last row dependent on 2 or more other rows

Give 0 marks if there is any evidence of a bubble sort.</td></tr>
<tr><td></td><td>(b)</td><td></td><td>• set of numbers broken into <u>multiple</u> sets
• uses pivots</td><td>[max 1]</td><td>[“multiple” essential]</td></tr>
<tr><td></td><td>(c)</td><td></td><td>• if stack is empty…
• …report error and stop
• output data(stack_pointer)
• decrement stack_pointer</td><td>[max 3]</td><td>Item in brackets shows position of data</td></tr>
<tr><td>6</td><td>(a)</td><td>(i)</td><td>• at the same level as Lorry…
• …as they are subclasses of Vehicle/they inherit properties from Vehicle</td><td>[max 2]</td><td>Accept diagram
Correct arrow to show inheritance for 2nd mark</td></tr>
<tr><td></td><td></td><td>(ii)</td><td>• Vehicle…
• …as it is the superclass/as other classes are subclasses/as all vehicles have the property</td><td>[max 2]</td><td></td></tr>
</table>

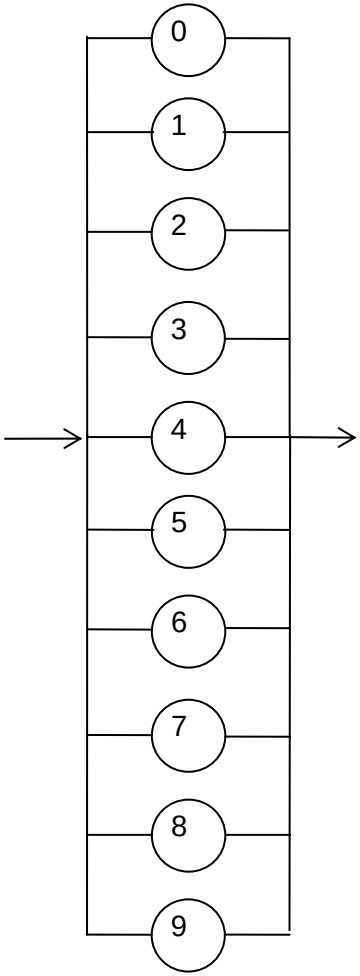
Question			Expected Answer	Mark	Additional Guidance
	(b)	**(i)**	• setPrice:/changeStatus:	**[1]**	
		(ii)	• model/engineType/price/status	**[1]**	
		(iii)	• Y	**[1]**	
	(c)	**(i)**	• state diagram	**[1]**	Accept “state” only
		(ii)	• entry point	**[1]**	Accept a description
		(iii)	• exit point	**[1]**	Accept a description
7	**(a)**	**(i)**	• to define terms unambiguously (for a computer language)	**[1]**	
		(ii)	• Backus-Naur Form	**[1]**	Accept BNF

Question	Expected Answer	Mark	Additional Guidance
(iii)			

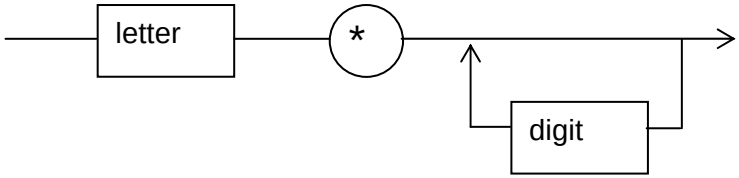
letter:



digit:



label:



Question			Expected Answer	Mark	Additional Guidance
		(iii) cont	*Award all 5 marks for a complete definition on single or multiple diagrams.* *Award partial marks as follows, 1 mark per bullet:* • correct use of circles, rectangles & arrows in diagrams • define letter A-E • define digit 0-9 • letter, *, digit in order with no extra terms • correct position & loop around digit	**[max 5]**	Accept correct solution with additional term (e.g. "number") defining a series of 0 or more digits
	(b)	**(i)**	• any expression can be processed in order (left to right) • no rules of precedence are needed/no brackets are needed/unambiguous	**[max 1]**	
		(ii)	*Full marks for pq+rs-** *If incorrect answer, award partial marks for:* • an expression starting with pq • pq+ anywhere in expression • rs- anywhere in expression	**[max 4]**	
		(iii)	• post order	**[1]**	

Question			Expected Answer	Mark	Additional Guidance
8	(a)		*[Points include the following, but should be written to make comparisons]* *Machine code* • written in binary/hexadecimal • no translation needed • very difficult to write *Assembly language* • includes mnemonics • includes names for data stores • translated by an assembler • easier to write than machine code, but more difficult than high level language	**[max 4]**	
	(b)		• address field (in an instruction) • it holds data… • to be used by the operation given in the opcode • eg in ADD 12, "12" is the operand *[accept any valid example]*	**[max 4]**	
	(c)		*direct:* • the simplest/most common method of addressing • uses the (data in) the address field… • …without modification • eg In ADD 23, use the number stored in address 23 for the instruction (accept any valid example) • limits the memory locations that can be addressed * *indirect:* • uses the address field as a vector/pointer… • …to the address to be used • used to access library routines • eg In ADD 23, if address 23 stores 45, address 45 holds the number to be used (accept any valid example) • increases the memory locations that can be addressed * *[max 4 for either, total max 6]*	**[max 6]**	* allow only one of these

<table>
<tr><th colspan="3">Question</th><th>Expected Answer</th><th>Mark</th><th>Additional Guidance</th></tr>
<tr><td>9</td><td></td><td></td><td>
<table>
<tr><th></th><th>DDL only</th><th>DML only</th><th>Both DDL & DML</th></tr>
<tr><td>It is used to create new tables.</td><td>✓</td><td></td><td></td></tr>
<tr><td>It defines foreign keys.</td><td>✓</td><td></td><td></td></tr>
<tr><td>It can query data.</td><td></td><td>✓</td><td></td></tr>
<tr><td>It can sort data into an order.</td><td></td><td>✓</td><td></td></tr>
<tr><td>It is used to update the data.</td><td></td><td>✓</td><td></td></tr>
<tr><td>It is a high level language.</td><td></td><td></td><td>✓</td></tr>
<tr><td>It is used for writing the schema.</td><td>✓</td><td></td><td></td></tr>
</table>
[1 mark per row, max 7]
</td><td>[max 7]</td><td></td></tr>
<tr><td>10</td><td>(a)</td><td>(i)</td><td>• many-many [accept E-R diagram]</td><td>[max 1]</td><td></td></tr>
<tr><td></td><td></td><td>(ii)</td><td>• not allowed/not in 3NF
• needs another table between Student & Subject…
• …to avoid duplication of data/to change to 3NF</td><td>[max 3]</td><td>Accept “not normalised” for “not in 3NF”</td></tr>
<tr><td></td><td>(b)</td><td>(i)</td><td>• many-one</td><td>[1]</td><td></td></tr>
<tr><td></td><td></td><td>(ii)</td><td>• both tables have their own primary key…
• …which is a unique identifier
• primary key from PersonalTutor…
• …is used as an attribute in Student…
• …is a foreign key in Student…
• …used to create the relationship/link between tables</td><td>[max 5]</td><td>3rd bullet must be qualified</td></tr>
</table>

Question			Expected Answer	Mark	Additional Guidance
	(c)		*Meaning:* • (an attribute that) can be used to search for a group of records… • … or allows records to be accessed in a different order *[max 1]* *Example:* • eg search for TutorId in Student to find all students with a particular personal tutor • eg StudentSurname in Student can access students in alphabetical order *[accept other relevant example]* *{max 1]*	**[max 2]**	
	(d)		• presentation of selected data… • …usually in the form of a table/specific layout • may be defined in advance… • …so the user does not need to set it up *[max 2]* *Features of report definition* • a query • a display order *[max 2]*	**[max 4]**	
			Total	**[120]**	

OCR (Oxford Cambridge and RSA Examinations)
1 Hills Road
Cambridge
CB1 2EU

OCR Customer Contact Centre

14 – 19 Qualifications (General)
Telephone: 01223 553998
Facsimile: 01223 552627
Email: general.qualifications@ocr.org.uk

www.ocr.org.uk

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored

Oxford Cambridge and RSA Examinations
is a Company Limited by Guarantee
Registered in England
Registered Office; 1 Hills Road, Cambridge, CB1 2EU
Registered Company Number: 3484466
OCR is an exempt Charity

OCR (Oxford Cambridge and RSA Examinations)
Head office
Telephone: 01223 552552
Facsimile: 01223 552553

© OCR 2011