OCR
Oxford Cambridge and RSA

GCE

Computing

Unit **F453:** Advanced Computing Theory

Advanced GCE

Mark Scheme for June 2017

OCR (Oxford Cambridge and RSA) is a leading UK awarding body, providing a wide range of qualifications to meet the needs of candidates of all ages and abilities. OCR qualifications include AS/A Levels, Diplomas, GCSEs, Cambridge Nationals, Cambridge Technicals, 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 2017

Question	Answer/Indicative content	Mark	Guidance
1 a (i)	• Spooling	**1**	CAO
1 a (ii)	• Processor is not waiting for the printer to respond • Can deal with other jobs/tasks • Avoids speed mismatch	**2**	1 per bullet, max 2
1 b (i)	• To allow programs that are too large to fit into RAM… • …to still be able to run	**2**	
1 b (ii)	• Uses a part of the backing store… • …as temporary memory • Holds part of the program not currently in use • Uses paging/segmentation • Pages/Segments swapped between main memory and backing store	**4**	1 per bullet, max 4

1 c	• Utility programs
	  • Programs that help with the efficient running of the OS…
	  • …not part of the OS
	• Device drivers
	  • Protocols/software…
	  • …that enable parts of the machine to communicate/communicate with other devices
	• User Interface
	  • Allows the user to interact with the hardware/applications
	  • So that the computer can do something useful
	• File Allocation Table
	  • Stores file names, file sizes, access rights, free space
	  • Is used by the OS when accessing files/pointers to the start of files
	  • Is updated by the OS when files are written/updated/deleted
	• Allow one example only for each explanation
		6	1 per named component, max 2 description bullets per component, max 6 total
			Not interact with O/S

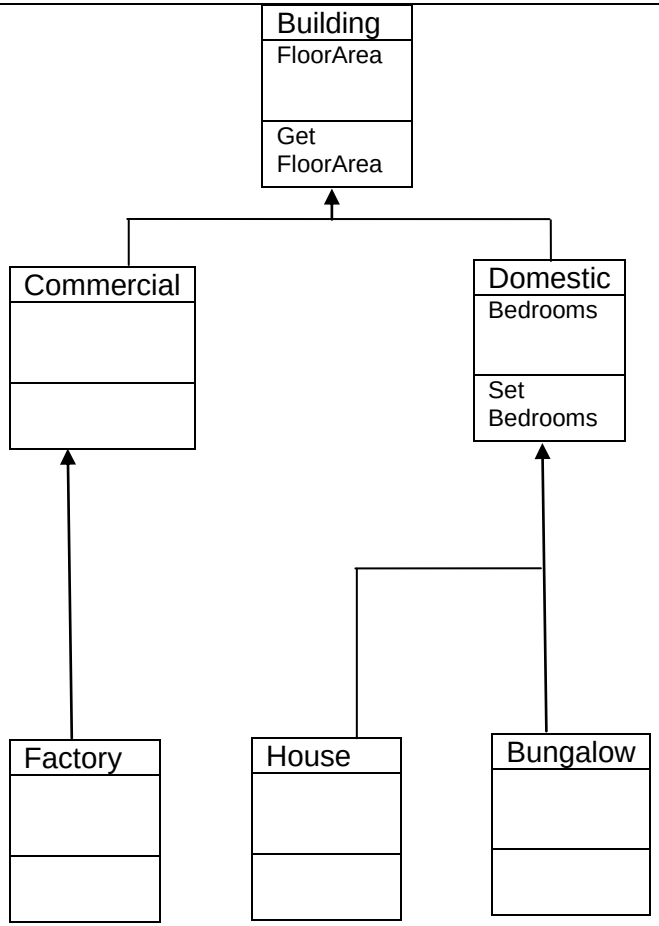
1 d	• advantage: Most urgent jobs/tasks get done first • disadvantage: low priority jobs/tasks may not get completed/ may take a long time to be completed	**2**	

2 a	**Mark band 6-8. High level response.** Candidate has given a comprehensive response stating most of the bullets for both points, made good comparisons between the two and has used appropriate technical language throughout their answer. There are few, if any, spelling or grammatical errors. **Mark band 3-5. Medium level response.** Candidate has given an adequate response stating some of the bullets for both points and made some comparisons between the two or has explained one comprehensively. The candidate has used some appropriate technical language in their answer. There may be a few spelling or grammatical errors. **Mark band 0-2. Low level response.** Candidate has given an adequate response stating some of the bullets for one or both points. The candidate may have used some appropriate technical language in their answer. There are some spelling or grammatical errors. • some statement about the fact that the interpreter will be used for testing/ compiler for finished product Relevant points to be made:- Compiler • Creates an executable file • Debugging is more difficult • Can produce spurious errors • Reports errors all at once • Can produce intermediate code • Easier to protect Copyright of program • Uses more memory • Can use Dynamic link libraries • Whole program needs to be loaded into memory	**8**	

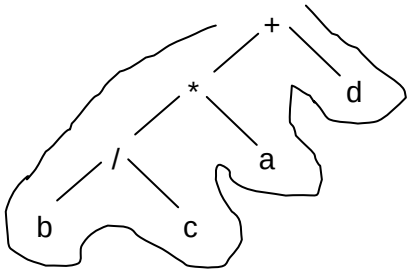
	Interpreter • Interprets one line at a time • Finds an error and stops straight away • Useful for designing/ testing a program • Uses less memory • Source code only needs to be loaded into memory one line at a time • Uses a virtual machine • Interpreter has to be in memory at the same time as the code		
2 b	• Creates tokens • A token is a sequence of characters/bits • Creates a symbol/lookup table • The symbol table holds details of variables (and subroutines) • Symbol table includes details such as variable name/datatype/scope • Lexical analysis passed onto syntax analysis • Unnecessary characters such as spaces or comments are removed • May report errors • Errors reported as a list	**4**	Allow 'Tokens are normally 16 bit' 1 per bullet, max 4
2 c	• The program will be parsed once to generate the code • Optimisation begins as a second parse is made • Looking for areas where code can be shortened • e.g. x = 10 y = x could be shortened to y = 10	**3**	Allow any reasonable example. 1 per bullet, max 3

3 a	• Receives address from the Program Counter (PC) • MAR used to <u>point to an address</u> in memory where data is to be fetched from • MDR used to fetch/hold data from the address pointed to by the MAR If writing to a memory address • MAR is passed an address by the CIR • Data from Acc is passed to MDR… //Acts as a gateway • …which places the data at the address pointed to by the MAR	**4**	1 per bullet, max 4
3 b	Similarities: • Both should speed up processing • Both have multiple ALU's • Both are difficult to program//Complex operating system Differences • Array has a single Control Unit • Parallel has multiple Control Units • Parallel can process different tasks simultaneously, whereas Array can effectively only process the same task simultaneously (SIMD) • Allow examples of either if relevant	**4**	1 per bullet
3 c	Advantages: • Faster when handling large sets of data with the same instruction • Not limited by the bus transfer rate Disadvantages: • More difficult to program • No benefit for processes that only require a single piece of data	**2**	
4 a (i)	• 011011 • 6	**2**	

<table>
<tr><td>4 a (ii)</td><td>• 100011
• -8</td><td>2</td><td></td></tr>
<tr><td>4 b (i)</td><td>• 0.011
• Move decimal point 0.11
• Exponent -1 = 111
• 01100111 Answer</td><td>4</td><td></td></tr>
<tr><td>4 b (ii)</td><td>• 01101 reverse numbers and add 1
• Exponent is 1
• -1.101
• -1.625 / -1 5/8</td><td>4</td><td>Allow for different methods of working out</td></tr>
<tr><td>5 a</td><td>• Check if print queue is empty
• If so report error and stop
• Else get item from <u>Front</u> pointer
• Increment pointer</td><td>4</td><td>Must be front pointer, just pointer TV</td></tr>
<tr><td>5 b</td><td>• Data files must be the same data type
• Each item needs to be checked (on both files)
• Duplicate/repeated items are allowed/disallowed
• If one file is empty then the rest of the other file is appended
• Files are checked to see if they are empty before starting
• If records have multiple fields, file must have a common key</td><td>3</td><td>1 per bullet, max 3</td></tr>
<tr><td>6 a</td><td>
<table>
<tr><td></td><td>Procedural</td><td>Object oriented</td><td>Declarative</td></tr>
<tr><td>Has facts</td><td></td><td></td><td>✓</td></tr>
<tr><td>Uses inheritance</td><td></td><td>✓</td><td></td></tr>
<tr><td>States how a problem is solved</td><td>✓</td><td>(✓)</td><td></td></tr>
<tr><td>Uses rules</td><td></td><td></td><td>✓</td></tr>
<tr><td>One line at a time in order</td><td>✓</td><td>(✓)</td><td></td></tr>
<tr><td>Uses methods</td><td></td><td>✓</td><td></td></tr>
</table>
</td><td>6</td><td>Allow either Procedural or Object oriented for ‘One line at a time in order’</td></tr>
</table>

6 b	 • Sub-classes of Commercial and Domestic • Sub-classes of House and Bungalow under Domestic and Factory under Commercial • Attribute FloorArea in Buildings • Attribute Bedrooms in Domestic • Methods both in the correct place • Arrows pointing from sub-class to super-class	6 9	

6 c (i)	• Activity	**1**	CAO
6 c (ii)	• A = Decision • C = Input/Output • D = End	**4**	
6 c (iii)	• Line covering more than one activity	**1**	
7 a (i)	• Stack is LIFO • Function is pushed onto the stack • When finished, function is popped off the stack • Answer can be in a diagram or written	**3**	
7 a (ii)	• Can be passed by value or by reference • Used to pass data into/ out of procedures/functions	**2**	
7 b (i)	• False • True • False • False	**4**	
7 b (ii)	• <number> ::= <digit> | <number><digit> • <code2> ::= <lowercase><number> • correct use of ::= • Correct use of <> |	**4**	Allow any reasonable name for new rules. <number><digit> in the example can be either way round

7 c	Tree diagram is good for showing working
	 
	
	• Mark for showing working (any method)
	• One mark for full answer
	  ○ bc/a*d+ a*bc/d+
	  ○ Alternate answer is dabc/*+
	• bc/ alternate dacb
	• a* d+ alternate /*+
		4	Accept other methods for working
			bc can be either way round in both alternatives

8 a (i)	• direct • indirect • immediate • indexed • relative • symbolic	**3**	1 per bullet, max 3

8 a (ii)	direct • refers to an (absolute) memory location • fast method of access indirect • address is at an address that holds the location • of the actual address immediate • fastest method of addressing • actual value is the one required • no need to access memory indexed • index adds an offset • to the base address • increments each time relative • a fixed offset is added • to the current value in the program counter symbolic • The use of characters • to represent the address of a store location	**4**	1 per bullet, max 2 per mode, max 4 total
8 b	Mnemonic • SUB is the mnemonic • A memorable name/easy to remember • given to a specific instruction Opcode • A single instruction • Specific to the machine architecture • indicates what to do Operand • 42 is the operand • The data that is operated on • by the Opcode • Holds data/address to be used	**6**	1 per bullet, max 2 per term, max 6 total

9 a (i)	• BookId • LibraryId	**2**	CAO
9 a (ii)	• LibraryId • Book table	**2**	
9 b (i)	• Data Description/Definition Language	**1**	CAO
9 b (ii)	• Data Manipulation Language	**1**	CAO
9 b (iii)	• Names of tables • Names of attributes/fields • Data types of fields • Descriptions of fields • Number of rows • Relationships between tables • Access rights • Usage statistics • Validation rules • Field length • Primary/Foreign keys • Indexes	**6**	

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

OCR Customer Contact Centre

Education and Learning
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 2017

PART OF THE CAMBRIDGE ASSESSMENT GROUP

LLOYD'S REGISTER QUALITY ASSURANCE ISO9001
UKAS MANAGEMENT SYSTEMS 001