AQA

ASSESSMENT and
QUALIFICATIONS
ALLIANCE

Mark scheme
January 2003

GCE

Computing

Unit CPT4

# Unit 4: Processing and Programming Techniques

The following notation is used in the mark scheme:
- ; means a single mark;
- / means alternative response;
- **A** means acceptable creditworthy answer;
- **R** means reject answer as not creditworthy;
- **I** means ignore;
- **BoD** means benefit of doubt

1 a i   Data must be in given order

	1	pheasant	2
	2	teal	3
	3	widgeon	5
START	4	partridge	1
	5	woodpigeon	0

//
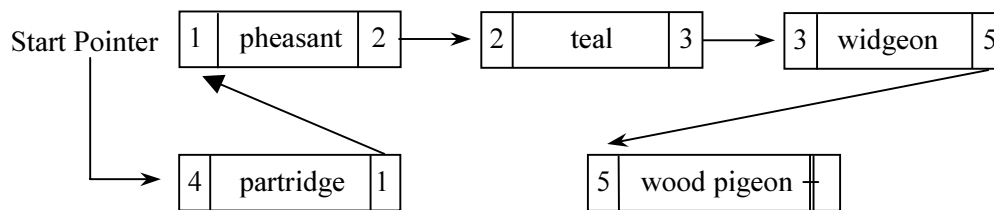
Start Pointer

End Pointer can be blank
1 for correct START and END pointers;
1 for correctly numbered nodes and correct pointers (need all birds);

ii

	1	pheasant	7
	2	teal	3
	3	widgeon	5
	4	partridge	1
	5	woodpigeon	0
START	6	grouse	4
	7	snipe	2

//   correctly amended diagram

1 for grouse and snipe physically at end;
1 for correct pointers (if not as ms than clear and logical);   **2**

b   The amount of memory taken up can vary;
     // The size / length of the structure / linked list can vary;   **1**

     at run time;   **1**

c   A heap / stack/ a pool of available locations;
     A pointer holds the address of the allocated block / next available location;   **2**   **8**

2	i	multi-tasking;	**1**	
	ii	multi-programming;	**1**	
	iii	multi-user;	**1**	**3**

3	a i	8;	**1**	
	a ii	Each time a comparison is made in a binary search the number of items to be searched / list is halved;	**1**	
		// 137 lies between 2^7 and 2^8;		
		Could give (ii) even if (i) incorrect		
	b i	137;	**1**	
	b ii	In a linear search of 137 items, the required item might be the 137th one;	**1**	
		need a termination – must explain why 137 is the <u>maximum</u>		**4**

In linear search 137 items — rendered: In a linear search of 137 items, the required item might be the 137th one;

4	a	command line interface;	**1**	
	b	command line interpreter / CLI;	**1**	
		A shell		
		c and d must be in correct context		
	c	user has access to all instructions – GUI can restrict access / can control hardware;		
		CLI takes up less memory than a GUI / uses less processing power / No powerful graphics card required;		
		Instructions execute faster than those given via a GUI / CLI is quicker to use for the experienced user than a GUI;		
		Any two	**2**	
	d	The user has to remember the precise format of each instruction;	**1**	**5**
		Not intuitive / user friendly; *(less intuitive needs a than')*		

5	a i	1011 1101 1001 0011;	**1**	
	ii	1011101000 000011		
		–ve number;	**1**	
		exponent +3; *(explained or demonstrated)*	**1**	
		value $4\,^3/_8$;	**1**	
		Answer $–4\,^3/_8$ / –4.375		
		1 mark for each of three points to max 3		
	b	Normalisation ensures the maximum possible accuracy for a given number of bits; *(given no. of bits can be implied – e.g. leading zeroes can be replaced by significant digits at the end of the mantissa)*		
		Arithmetic operations simplified		
		Ensures that only a single representation of a number is possible;		
		Any 2 points to max	**2**	**6**

| 6 | a | add; uses the rules 0+0=0, 0+1=1, 1+1=10; | **1** | |
		AND; uses the rules 0 AND 0 =0, 0 AND 1 = 0, 1 AND 1 = 1;	**1**	
		// AND operates on a bit by bit basis;	**1**	
		ADD carries out arithmetic between bits;	**1**	
		// ADD performs addition, AND compares bits;	**1**	
		AND outputs a 1 if both inputs are 1, otherwise 0;	**1**	
	b i	A register / the accumulator;	**1**	
		A general purpose register		
		R the wrong register e.g. MDR		
	ii	Access to main memory is slower than to a register;		
		Would need to write results to MM and read them back again for each		
		instruction;	**2**	**5**

7	a	chris, clauses 10, **8** and 2; *at least 2 clauses needed*	**1**	
		alan clauses 11,4 and 1; *at least 2 clauses needed*	**1**	
		names only, none,		
	b	cousin (X,Y) IF grandfather (Z,X) AND grandfather (Z,Y);;;	**3**	
		overall structure 'cousin(X,Y) IF grandfather () AND grandfather()' 1		
		(W,X) 1		
		(W,Y) 1		
		alternative approach mark in similar mode		
	c	Processing of natural language;		
		Medical diagnosis;		
		Image interpretation;		
		Other valid types;		
		1 mark for each of 2 valid suggestions to max	**2**	**7**
		A analysing family tree, Artificial Intelligence;		

8	a i	the operand field contains the data required for the operation;	**1**	
	ii	the operand refers to a location in main memory;	**1**	
		which holds the data required;	**1**	
	iii	the operand address is calculated by adding the value given;	**1**	
		to the contents of an index register;	**1**	
	b i	0 5;	**1**	
	ii	120 5;	**1**	
	ii	121 5;	**1**	
	iv	127 5;	**1**	**9**
		−1 mark if only changes filled in		

9	a	Files are stored in separate directories / folders; **A** path / pathway	**1**	
	b	disk stores a disk map to indicate free and used blocks / FAT; disk stores information such as disk volume name / label/ capacity of disk / number of sectors; boot sector / messages; disk stores directories for disk / contains system files; Any 2 points @ 1 each	**2**	
	c	*n.b. loading executable file*		
	i	searches directory for match with given file name; obtains a block address / uses FAT to locate file; obtains size of file; load file into memory; checks file is compatible / not corrupted / no virus; check file status (not already open, licensed) *Any 2 @ 1 each* *Error*: no matching file / invalid file name / file corrupt / file already opened / incompatible file type;	**2** **1**	
	ii	finds required amount of memory space / allocates memory; loads into memory / co-ordinates sharing of memory (DLLs); relocating loader translates addresses; using size of file; marks memory as taken / keeps track of where loaded; allocates base and limit register values; set up Page Management Table / **A** allocates segment descriptor table; manages virtual memory; *Any 3 @ 1 each* *Error*: insufficient memory / memory full.	**3** **1**	**10**

10 a Produces re-usable code because of inheritance / encapsulation;
Data is only accessible in well defined ways (because encapsulated);
More efficient to write programs which uses pre-defined / inherited objects /
classes;
*Storage structure of data and the code in an object may be altered without
affecting programs that make use of the object;*
Code produced contains fewer errors / more reliable;
Solutions are easier to understand when expressed in terms of objects;
Easier to enforce design consistency – Windows GUI functionality;
*Cheaper production costs / Less maintenance effort required by developer since
reliable 'objects' can be re-used / bought in;*
New functions can be added to objects easily (because encapsulated);
Any 2 advantages @ 1 each – must state an advantage, not make a statement.   **2**
R Object is independent.

b
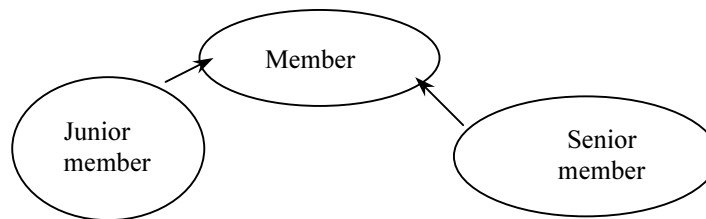
1 mark for correct base class and derived classes *;*   **1**
1 for two correctly directed arrows;   **1**

c   Member = **Class**
              (Procedure) AddNewMembers;   }
              (Procedure) AmendMembers;   }   ;   **1**
              (Procedure)ShowMembers;   }
              Private   ;   **1**
                  MembershipNo : Integer
                  Name : String;
                  Address : String;   ;;   **2**   **8**
              End;

Exact syntax not required, but must be in style of .
3 procedures   1
Private   1
All 3 field (property) names   1
3 reasonable data types   1