Version 1.0

General Certificate of Education
June 2010

Computing COMP3

Unit 3 Problem Solving, Programming, Operating Systems, Databases and Networking

Final

Mark Scheme

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.

Notation used in mark scheme:

; - means a single mark
// - means alternative response
/ - means an alternative word or sub-phrase
A - means acceptable creditworthy answer
R - means reject answer as not creditworthy
I - means ignore
DPT - don't penalise twice. A mistake that might otherwise result in the loss of more than one mark should only lose the first mark. Subsequently the mistake should be ignored.

1	**(a)**		Can be used over longer distances // fewer wires/cables/lines required // only one pathway required // cheaper to cable // no risk of data skew // easier to repeat/regenerate/switch // switching equipment/repeater design is easier/cheaper // no crosstalk; **R** Cheaper **NE** **R** Cannot get out of synch	1

1	**(b)**		*Parity Bit:* 1; *Start bit, Stop Bit :* Can be either 0 or 1, but must both be different to get mark;	2

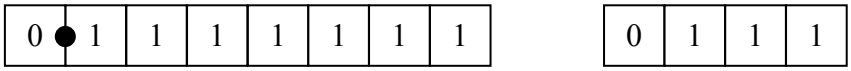
1	**(c)**		*Definition (1 mark):* Receiver and transmitter (clocks) do not need to be/are not (exactly) synchronised // transmission of data without use of external clock signal // receiver and transmitter clock only synchronised at start of/for length of transmission // data sent as soon as available rather than waiting for clock pulse/ synchronisation symbol; *Explanation of start and stop bits (2 marks):* Start bit synchronises receiver (clock) (to transmitter/data) // locks receiver and transmitter in phase // starts receiver's clock // wakes receiver; Stop bit allows start bit to be recognised // allows receiver to process received bits; Start and stop bits indicate when data is being transmitted/begins; **MAX 2**	3

2	**(a)**		To hide the complexities of the hardware from the user // Provision of virtual machine; **R** Provision of user interface NE **A** machine for hardware but **R** system, computer Manages the hardware (resources) // allocation of hardware resources (to processes); **A** Examples of resources	2

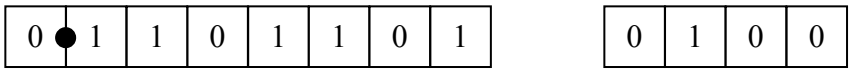
2	**(b)**	**Subject-related points:** *Desktop computer used for many different/generic purposes; so …* • desktop OS must support a wide range of peripherals/storage devices; • desktop OS must run wide range of software/packages; • desktop OS more customisable (by user); • desktop OS must manage security; *Desktop computer requirements more likely to change over time // more likely to want to add new features/support new applications; so....* • desktop OS has modular design / easier to upgrade; *Desktop computers made by many different manufacturers // to varying specifications; so....* • desktop OS must run on wider range of hardware platforms; *Desktop computer more likely to be networked; so…* • desktop OS must support networking protocols; *Embedded systems (often) made at low cost // may have minimal processing requirement; so…* • embedded system OS has lower hardware requirements (allow e.g. such as slower processor, less RAM); *Embedded systems have few inputs and outputs to user; so…* • embedded system OS provides no/minimal user interface; • embedded system OS designed to deal with input from sensors // output to control devices; *Embedded systems (often) in battery powered devices; so…* • managing power consumption particularly important; **A** example of power management *Embedded systems (are sometimes) real-time // for safety-critical applications; so…* • real-time embedded system OS must be designed to guarantee speed of response // respond very quickly;* • real-time embedded system OS must deal with many inputs simultaneously;* • real-time embedded system OS may need to be failsafe;* • real-time embedded system OS may incorporate redundancy;* ** - These points only valid if real-time system specifically referenced.* Accept any mix of points. The reason (in italics) does not have to be stated to award a mark for the difference/feature. Accept converse of the points if the point itself has not been given.	

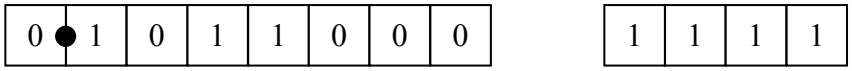
Mark Bands and Description	
3-4	*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).* *SUB* Candidate has made three or more relevant points. *QWC1* Text is legible. *QWC2* There are few, if any, errors of spelling, punctuation and grammar. Meaning is clear. *QWC3* The candidate has selected and used a form and style of writing appropriate to the purpose and has expressed ideas clearly and fluently. *QWC4* Sentences and paragraphs follow on from one another clearly and coherently. *QWC5* Appropriate specialist vocabulary has been used.
2	*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).* *SUB* Candidate has made two relevant points. *QWC1* Text is legible. *QWC2* There may be occasional errors of spelling, punctuation and grammar. Meaning is clear. *QWC3* The candidate has, in the main, used a form and style of writing appropriate to the purpose, with occasional lapses. The candidate has expressed ideas clearly and reasonably fluently. *QWC4* The candidate has used well-linked sentences and paragraphs. *QWC5* Appropriate specialist vocabulary has been used.
1	*To achieve a mark in this band, candidates must meet the subject criterion (SUB). The quality of written communication should be typified by the QWCx statements.* *SUB* Candidate has made just one relevant point. *QWC1* Most of the text is legible. *QWC2* There may be some errors of spelling, punctuation and grammar but it should still be possible to understand most of the response. *QWC3* The candidate has used a form and style of writing which has many deficiencies. Ideas are not always clearly expressed. *QWC4* Sentences and paragraphs may not always be well-connected or bullet points may have been used. *QWC5* Specialist vocabulary has been used inappropriately or not at all.
0	Candidate has made no relevant points.

			Note: Even if English is perfect, candidates can only get marks for the points made at the top of the mark scheme for this question. **If a candidate meets the subject criterion in a band but does not meet the quality of written communication criteria then drop mark by one band, providing that at least 3 of the quality of written communication criteria are met in the lower band. If 3 criteria are not met then drop by two bands.**	4

3	**(a)**		 **1 mark** for correct bit pattern in both mantissa and exponent.	1

3	**(b)**		Mantissa = -0.6875 // -11/16 Exponent = 3 Answer = -5.5 // -5½ **1 method mark** for either: • showing correct value of both mantissa and exponent in denary • showing binary point shifted 3 places to right within a correct binary pattern* • indicating that final answer calculated using answer = mantissa x $2^{exponent}$ (**A** mantissa in denary or binary but exponent must be in denary) **1 mark** for correct answer * Correct binary patterns with the binary point shifted 3 places are: 1010.1000 0101.1000 1010.1 101.1000 101.1	2

3	**(c)**		 **1 mark** for correct mantissa **1 mark** for correct exponent	2

3	**(d)**		 **1 mark** for correct mantissa **1 mark** for correct exponent	2

3	**(e)**		*Definition (2 marks):* The result of a calculation is too large to store/represent // a number is too large to store/represent; In the available number of bits / storage space (allow example e.g. data type, byte, word, example of a data type); **R** space NE *Example (1 mark):* Multiplying two numbers together; Dividing a number by a number less than one / small number; **R** zero **A** Adding two numbers (of same sign) **A** When number converted from one type to another that does not have suitable range/enough bits/enough storage space to represent it **A** Answers by example **MAX 1**	3

4	**(a)**		<table><tr><th>Real number</th><th>Yes/No</th></tr><tr><td>203.412</td><td>Yes</td></tr><tr><td>-12.87</td><td>No</td></tr><tr><td>12.43E-12</td><td>Yes</td></tr></table> **1 mark** per correct Yes/No **A** other indicators that clearly mean Yes/No e.g. True/False, Tick/Cross.	3

4	**(b)**		<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 <whole-number> ::= <digit> | <digit> <whole-number> <integer> ::= <whole-number> | + <whole-number> | - <whole-number> **1 mark** for each correct rule Alternative for integer (1 mark, accept in either order): <symbol> ::= + | - <integer> ::= <whole-number> | <symbol> <whole-number> **A** <whole-number> defined with recursion other way around, i.e. <whole-number> ::= <digit> | <whole-number> <digit> **A** non-terminal names e.g. digit not enclosed in <> signs **A** spaces in non-terminal names e.g. whole number **A** terminal names enclosed in quotation marks e.g. “0”, ‘0’. **A** any sensible symbol for assignment e.g. ←, :=, =, : **A** ; as end-of rule marker; **A** any type of slash e.g. / for alternatives but **R** "or" **A** use of EBNF extensions for repetition and optional terms: <whole-number> ::= <digit> { <digit> } <integer > ::= [+ | -] <whole-number> **A** () for [] but **R** {} **R** rules that have additional options e.g. more than ten digits	

			DPT addition of chevrons or other symbols such as brackets to terminal symbols/rules unless they make meaning unclear	3

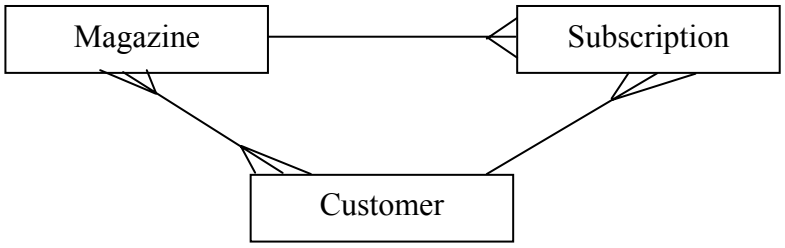
5	**(a)**	**(i)**	O(a^n); **A** exponential, a^n	1

5	**(a)**	**(ii)**	A;	1

5	**(b)**	**(i)**	The problem can be solved // algorithm exists for problem; But it cannot be solved in polynomial time // but not quickly enough to be useful; It takes an unreasonable amount of time; to solve; **A** too long time but **R** long time	2

5	**(b)**	**(ii)**	Use of heuristic; algorithm that makes a guess based on experience; That provides a close-to-optimal solution/approximation; that only works in some cases; **A** non-optimal Example of heuristic method e.g. hill-climbing/stochastic/local improvement/greedy algorithms/simulated annealing/trial and error/any reasonable example; Relax some of the constraints on the solution; **A** solve simpler version of problem **MAX 2**	2

6	**(a)**		**What means:** every attribute (in relation) is dependent on the key; the whole key and nothing but the key; OR (relations) contain no repeating groups (of attributes) // data is atomic; no partial dependencies; no non-key dependencies; **R** No repeated columns/attributes/data OR every determinant (in the relation) is a candidate key;; **MAX 2** **Why important:** Eliminate update anomalies; **A** Example **R** Easy to update **NE** Eliminate insertion anomalies; **A** Example Eliminate deletion anomalies; **A** Example Eliminate data inconsistency // improve consistency // avoid inconsistency problems; Minimise data duplication; **A** Reduce for minimise **R** elminate Eliminate data redundancy; **A** Reduce/minimise for eliminate **A** No unnecessarily repeated data **R** No repeated data **R** Saving space/memory **NE** **MAX 2**	4

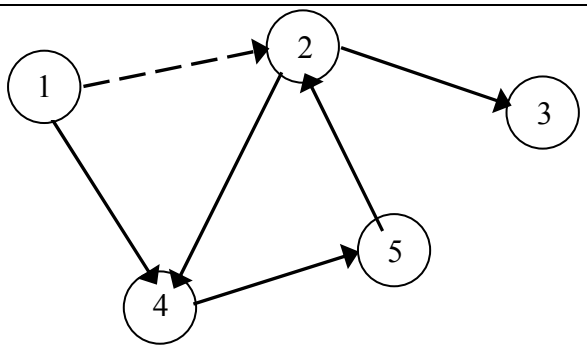
6	**(b)**		 1 mark for per correct relationship **MAX 2** **I** incorrect relationships	2

6	**(c)**		MagazineID VARCHAR(8) PRIMARY KEY(NOT NULL) // MagazineID VARCHAR(8) PRIMARY KEY(MagazineID) MagazineName VARCHAR(10) Category VARCHAR(10) SubscriptionRate SMALLMONEY Optional **1 mark** for MagazineID, and identified as primary key **1 mark** for other fields with appropriate data types **A** any sensible types / field lengths. Some examples are: For MagazineID: integer For SubscriptionRate: money, currency, float, real, decimal, dec, double, double precision, numeric As alternative to varchar: char, varchar, text, nchar, nvarchar, ntext, longvarchar, varchar2, nvarchar2 **R** answers clearly written in a different programming language	2

6	**(d)**		SELECT CustomerName, Address, Postcode FROM Magazine, Subscription, Customer WHERE MagazineName = 'AQA Computing Now' AND EndDate < '01/06/2010' AND Magazine.MagazineID = Subscription.MagazineID AND Subscription.CustomerID = Customer.CustomerID **1 mark** for SELECT clause with correct three fields (allow any additional fields from relations or *) **1 mark** for FROM clause with correct three tables **1 mark** for MagazineName = 'AQA ComputingNow' **1 mark** for EndDate < '01/06/2010' **1 mark** for two clauses linking tables on the common field **MAX 1** of the 3 marks for conditions if not joined by ANDs --- OR --- SELECT CustomerName, Address, Postcode FROM Magazine INNERJOIN Subscription ON Magazine.MagazineID=Subscription.MagazineID INNERJOIN Customer ON Subscription.CustomerID=Customer.CustomerID WHERE MagazineName = 'AQA ComputingNow' AND EndDate < '01/06/2010' **1 mark** for SELECT clause including correct three fields (allow any additional fields from relations or *) **1 mark** for correctly joining two tables in FROM clause **1 mark** for correctly joining the third table in FROM clause **1 mark** for MagazineName = 'AQA ComputingNow' **1 mark** for EndDate < '01/06/2010' **In both solutions:** Do not award mark for 'AQA Computing Now' unless it is enclosed in single or double quotation marks. For EndDate, accept # symbols or no delimiting symbols. Accept EndDate day and month without preceding 0, i.e. 1/6. Accept <= '31/05/2010' for EndDate. Accept table names before fieldnames. Accept use of Alias/AS command e.g. FROM Magazine as M then use of M as table name. Accept insertion of spaces into fieldnames **DPT** for unnecessary punctuation – allow one semicolon at the very end of the statement, but not at the end of each clause. **DPT** for fieldname before table name.	5

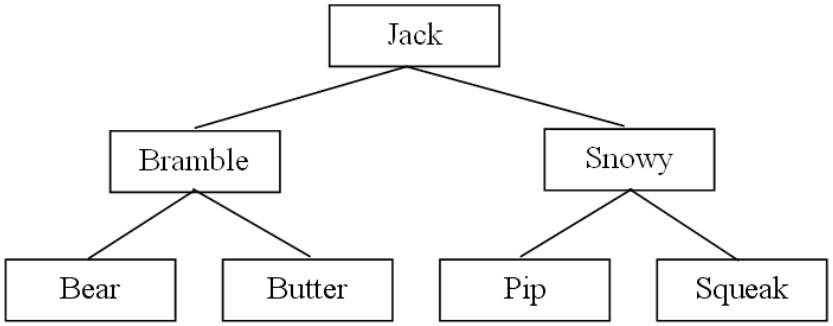
6	**(e)**		UPDATE Magazine SET MagazineName= 'AQA Garden News' WHERE MagazineName= 'AQA Gardening Monthly' **1 mark** per correct line **A** double or single quotes around magazine names **R** no quotes Accept table names before fieldnames. **DPT** for fieldname before table name.	

			MAX 2	2

7	**(a)**		 **1 mark** for all 5 lines correctly drawn **1 mark** for all 5 arrowheads pointing in correct directions **A** arrowheads at any position on line **MAX 1** if more than 5 lines drawn by candidate (note that dotted arrow is given in question)	2

7	**(b)**		Adjacency matrix appropriate when there are many edges between vertices // when edges may be frequently changed // when presence/absence of specific edges needs to be tested (frequently) Adjacency list appropriate when there are few edges between vertices // when graph is sparse // when edges rarely changed //when presence/absence of specific edges does not need to be tested (frequently) **A** alternative words which describe edge e.g. connection, line	2

7	**(c)**		*Connected* // There is a path between each pair of vertices; *Undirected* // No direction is associated with each edge; *Has no cycles* // No (simple) circuits // No closed chains // No closed paths in which all the edges are different and all the intermediate vertices are different // No route from a vertex back to itself that doesn't use an edge more than once or visit an intermediate vertex more than once; **MAX 2** **Alternative definitions:** Graph with no cycles, and a simple cycle is formed if any edge is added to it;; Graph which is connected, and it is not connected anymore if any edge is removed from it;; Graph in which any two vertices can be connected by a unique simple path;; (Note: not just adjacent vertices) Graph which is connected and has n − 1 edges where n is no of vertices;; Graph which has no simple cycles and has n − 1 edges where n is no of vertices;;	2

7	**(d)**		 **1 mark** for Jack as root **1 mark** for Bramble and Snowy as children of Jack **1 mark** for four correct children of Bramble and Snowy **DPT** if arrows drawn instead of lines **DPT** if any node has more than 2 child nodes Accept "mirror image" answers which are consistent.	3

7	**(e)**		**For solution with 3 arrays:** One array stores data items; One array for left child pointers; One array for right child pointers; Pointers stored at same location in arrays as corresponding data item; **For solution with 1 array of records:** Record created to store data item and pointers; One pointer to left child; One pointer to right child; **For either of the above solutions:** Rogue value (allow example) used to indicate no child; Variable indicates position in array(s) of root node // Root node stored at first location/start of array(s); **If answered as diagram:** Column for data with at least three correct data items in it; Use of rogue value for a node that does not have child; Correct value for a start pointer variable indicating position of root node in the array (not drawn as an arrow, array indices must be labelled); Column for left child pointers*; Column for right child pointers*; * = To get these marks, there must be a sufficient number of pointers to demonstrate that the data structure is a representation of a binary tree, but it is not necessary for every item to be shown. Also the array indices must be shown. **MAX 3**	3

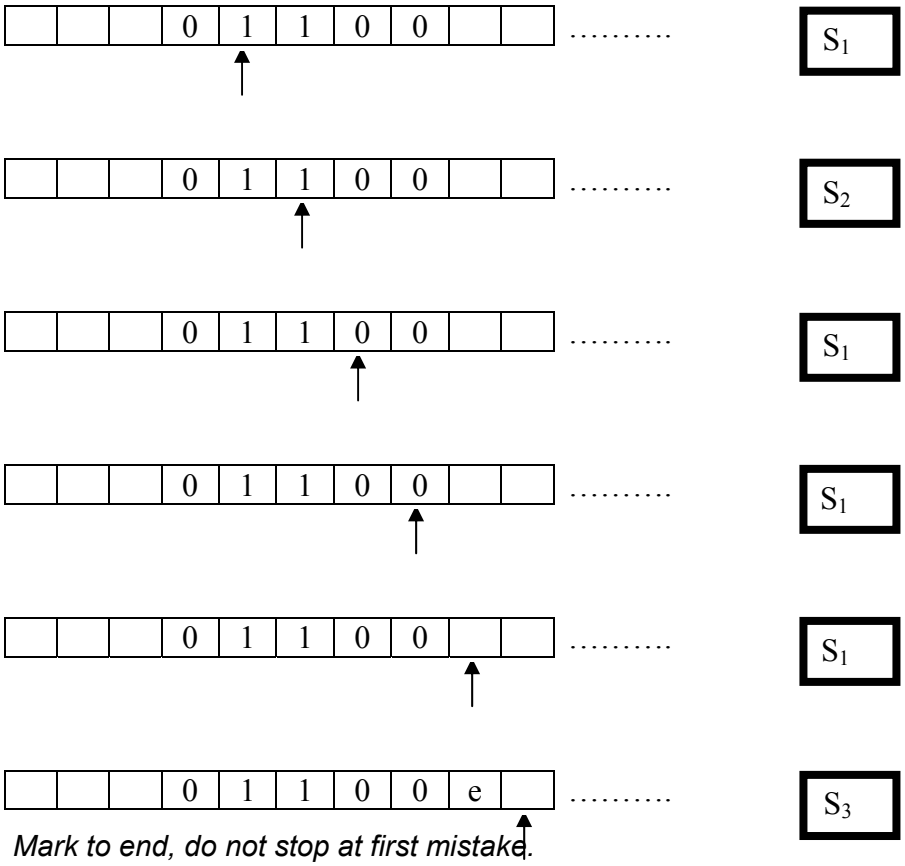
8 **(a)**

Number	Correct Label
❶	(0, 0, →)
❷	S_1
❸	(□, ○, →)
❹	S_3

1 mark for 1 and 3 correct – brackets not required
1 mark for 2 and 4 correct

2

8 **(b)**



Mark to end, do not stop at first mistake.
1 mark for first row correct;
1 mark for second row correct;
1 mark for both rows three and four correct;
1 mark for both rows five and six correct;
Must have correct tape contents and current state for each mark
A answers where the tape has been shifted
DPT for missing read/write head

4

8	**(c)**	Check if the tape contains an even / odd number of 1s // check parity of number on tape;	1

8	**(d)**	Turing machines provide a (general/formal) model of computation; Provides a definition of what is computable // a task is computable if (and only if) it can be computed by a Turing machine; No computing device can be more powerful than a Turing machine // any algorithm that can be computed by any computer can be computed by a Turing machine; (The Church–Turing thesis states that) if an algorithm exists then there is an equivalent Turing machine for that algorithm // a Turing machine that can implement the algorithm; Through the Halting Problem, can be used to prove that some functions cannot be computed; **MAX 2**	2

9	**(a)**	**(i)**	192.168.0.x where x is not 0 or 255;	1

9	**(a)**	**(ii)**	192.168.2.x where x is not 0 or 255	1

9	**(a)**	**(iii)**	192.168.2.y where y is not 0 or 255 and the IP address is different to the one in (ii)	1

9	**(b)**	Star;	1

9	**(c)**	Identify which other computers are on same segment // can have packets/data sent <u>directly</u> to them; Identify which other computers are on a different segment // must have packets/data sent to them via the router; **R** network for subnet **MAX 1** 255.255.255.0 / FFFFFF00 / 11111111111111111111111100000000;	2

9	**(d)**	Use of WEP/Wired Equivalent Privacy/WPA/WiFi Protected Access; (Strong) encryption of transmitted data; **R** encoding User/computer must enter/send a passphrase/certificate at start of communication before laptop allowed to connect; **A** key for passphrase **A** only allow password if used in correct context; Access point checks MAC/hardware address of laptop and only allows computers with a MAC/hardware address in a list of approved addressed to connect; **R** IP address Disable broadcast of SSID/identity; Reduce / limit power of transmitter; **MAX 2**	2

9	**(e)**	**Subject-related points:** (Applies to) bus (topology); Computer monitors/listens to (data signal on cable/bus); If (data) signal present // if cable/bus busy continue to wait; When no (data) signal present // when cable/bus idle start to transmit; Whilst transmitting, computer monitors cable/bus to check for collision // to check if signal is identical to what it is sending; Collision occurs if two computers (start) sending at same time // if two packets/frames in transit at same time; If collision detected, jamming signal/signal warning of collision sent; To ensure other (transmitting) computers aware of problem // to stop other computers sending data; Computer that detected collision also stops sending data; Then waits a random period before attempting to retransmit/repeating transmission/this process; Period is random to reduce likelihood of collision recurring (between computers that caused collision); If a collision occurs again then waits a longer random time before attempting to transmit again; Use of exponential back-off algorithm to determine wait time;	

Mark Bands and Description	
5-6	*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).* *SUB* Candidate has produced a detailed description of how CSMA/CD works, including what happens if there is a collision (at least 5 points). *QWC1* Text is legible. *QWC2* There are few, if any, errors of spelling, punctuation and grammar. Meaning is clear. *QWC3* The candidate has selected and used a form and style of writing appropriate to the purpose and has expressed ideas clearly and fluently. *QWC4* Sentences and paragraphs follow on from one another clearly and coherently. *QWC5* Appropriate specialist vocabulary has been used.
3-4	*To achieve a mark in this band, candidates must meet the subject criterion (SUB) and 4 of the 5 quality of written communication criteria (QWCx).* *SUB* Candidate has produced a reasonable description which may or may not cover what happens in the event of a collision (at least 3 points). *QWC1* Text is legible. *QWC2* There may be occasional errors of spelling, punctuation and grammar. Meaning is clear. *QWC3* The candidate has, in the main, used a form and style of writing appropriate to the purpose, with occasional lapses. The candidate has expressed ideas clearly and reasonably fluently. *QWC4* The candidate has used well-linked sentences and paragraphs. *QWC5* Appropriate specialist vocabulary has been

			used.	

Mark Bands and Description	
1-2	*To achieve a mark in this band, candidates must meet the subject criterion (SUB). The quality of written communication should be typified by the QWCx statements.* *SUB* Candidate has produced a very limited or unclear description of how CSMA/CD works. *QWC1* Most of the text is legible. *QWC2* There may be some errors of spelling, punctuation and grammar but it should still be possible to understand most of the response. *QWC3* The candidate has used a form and style of writing which has many deficiencies. Ideas are not always clearly expressed. *QWC4* Sentences and paragraphs may not always be well-connected or bullet points may have been used. *QWC5* Specialist vocabulary has been used inappropriately or not at all.
0	Candidate has made no relevant points.

Note: Even if English is perfect, candidates can only get marks for the points made at the top of the mark scheme for this question.

If a candidate meets the subject criterion in a band but does not meet the quality of written communication criteria then drop mark by one band, providing that at least 3 of the quality of written communication criteria are met in the lower band. If 3 criteria are not met then drop by two bands.

6

10 **(a)**

				List				
ListLength	New	P	q	[1]	[2]	[3]	[4]	[5]
4	*38*	-	-	*9*	*21*	*49*	*107*	
		1						
		2						
		3						
			4					107
			3				49	
						38		
5								

4,5 in sequence for ListLength;
1,2,3 in sequence for p;
4,3 in sequence for q;
Final list in array is 9, 21, 38, 49, 107;
Do not award a mark if additional values indicated e.g. 4 for p

4

10 **(b)**

Inserts an item/variable New into list at <u>correct position/preserving order//into sorted list (or equivalent)</u>;

1

10 **(c)** **(i)**

Static structures have fixed (maximum) size whereas size of dynamic structures can change // Size of static structure fixed at compile-time whereas size of dynamic structure can change at run-time;
Static structures can waste storage space/memory if the number of data items stored is small relative to the size of the structure whereas dynamic structures only take up the amount of storage space required for the actual data;
Dynamic data structures (typically) require memory to store pointer(s) to the next item(s) which static structures do not need;
MAX 1
A just one side of points, other side is by implication

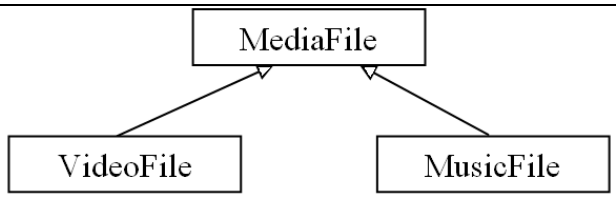
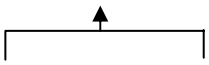
1

10 **(c)** **(ii)**

Heap is pool of free/unused/available memory;
Memory allocated/deallocated at run-time (to dynamic data structure);
MAX 1

1

11	**(a)**	A class/subclass has/shares/inherits properties and methods with the (parent) class (it is derived from); **A** another class Building a hierarchy of classes with each child class inheriting access to its parent class's methods and properties; Relationship between two object types/objects in which one object (type) is a kind of the other; **MAX 1** **A** Just one of properties and methods, do not need both. **A** The following as alternatives to properties: fields, attributes, characteristics, data with data as BOD **A** The following as alternatives to methods: procedures, functions, code. **A** The following as alternatives to parent: base, super. **A** The following as alternative to child: descendent, subclass, derived.	1

11	**(b)**	 **1 mark** for class names in boxes, with MediaFile drawn above the other two; **1 mark** for correct arrows; **A** arrows drawn as:  **A** filled/empty arrowheads **A** rotated through 90 degrees	2

11	**(c)**	Method can be defined with same name; **A** method can be redefined, an inherited method (but not just inheritance) as implying same name But have different implementation/code // perform different function; The redefined method will be used instead of the parent's method; **A** This is an example of polymorphism **A** Procedure, function, subroutine for method.	2

11	**(d)**	MusicFile = Class/Subclass (MediaFile) 1 Public Procedure PlayFile (Override) 1 Function GetArtist Function GetSampleRate } 1 Function GetBitDepth Private Artist : String SampleRate : Real } 1 BitDepth : Integer End **1 mark** for correct header including name of class and parent class; **1 mark** for redefining the PlayFile procedure; **1 mark** for defining all 3 extra functions needed to read variable values; **1 mark** for defining all 3 extra properties, with appropriate data types in private section; **A** any numeric types for SampleRate and BitDepth **A** answers that indicate separately that each variable is private **DPT** if any extra functions/procedures/variables included but do not penalise answers that have extra procedures to set variable values. **DPT** if any of the functions/procedures are private **I** parameters to methods, minor changes to names that do not affect clarity, case **-- OR --** (Public) class/subclass MusicFile extends/inherits MediaFile { 1 public void PlayFile (Override) 1 public string GetArtist() public float GetSampleRate() } 1 public int GetBitDepth() private string Artist private float SampleRate } 1 private int BitDepth } **1 mark** for correct header including name of class and parent class; **1 mark** for redefining the PlayFile procedure; **1 mark** for defining all 3 extra functions needed to read variable values; **1 mark** for defining all 3 extra properties, with appropriate data types as private; **A** any numeric types for SampleRate and BitDepth **DPT** if any extra functions/procedures/variables included but do not penalise answers that have extra procedures to set variable values. **DPT** if any of the functions/procedures are private **I** parameters to methods, minor changes to names that do not affect clarity, case **-- ACCEPT MIXES OF TWO METHODS IF MEANING IS CLEAR –**	4