National
Qualifications
2016

Computing Science

Advanced Higher

Finalised Marking Instructions

General Marking Principles for Advanced Higher Computing Science

This information is provided to help you understand the general principles you must apply when marking candidate responses to questions in this paper. These principles must be read in conjunction with the detailed marking instructions, which identify the key features required in candidate responses.

(a) Marks for each candidate response must always be assigned in line with these General Marking Principles and the Detailed Marking Instructions for this assessment.

(b) Marking should always be positive. This means that, for each candidate response, marks are accumulated for the demonstration of relevant skills, knowledge and understanding: they are not deducted from a maximum on the basis of errors or omissions.

(c) If a specific candidate response does not seem to be covered by either the principles or detailed Marking Instructions, and you are uncertain how to assess it, you must seek guidance from your Team Leader.

(d) Marks should be awarded regardless of spelling as long as the meaning is unambiguous.

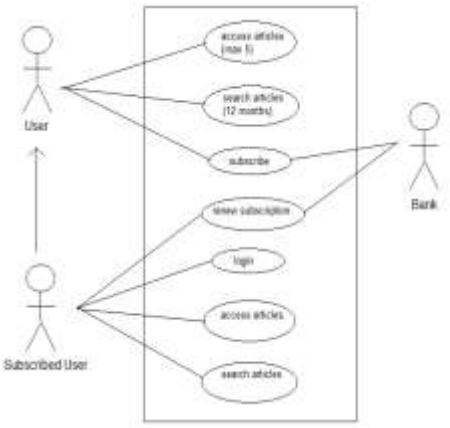
(e) Candidates may answer programming questions in any appropriate programming language or pseudocode. Marks should be awarded, regardless of minor syntax errors, as long as the intention of the coding is clear.

(f) Where a question asks the candidate to **describe**, the candidate must provide a statement or structure of characteristics and/or features. This should be more than an outline or a list. It may refer to, for instance, a concept, process, experiment, situation or facts in the context of, and appropriate to, the question. The candidates will normally be required to make the same number of factual/appropriate points as there are marks available for the question.

(g) Where a question asks the candidate to **explain**, marks should only be awarded where the candidate goes beyond a description, for example by giving a reason, or relating cause to effect, or providing a relationship between two aspects. These will be related to the context of the question or a specific area within a question.

(h) Credit should be given where a labelled diagram conveys clearly and correctly the response required by the question.

Detailed Marking Instructions for each question

Question			Expected Answer(s)	Max Mark	Additional Guidance
1.	(a)			**3**	1 mark for user and subscribed user actors 1 mark for 7 bulleted use cases 1 mark for any additional feature of a use case diagram eg associations (include, extend, inherit), actor representing payment external system, etc.
	(b)	(i)	Answers relating to the appropriate nature of the interface design in relation to user experience of the website when accessed from user's smartphone or tablet. *Accuracy of information entered* Both interfaces have a topic option which requires text input; this provides flexibility in terms of topic to be searched but potential for misspelling of topic exists. Selection of data from a small calendar on a smartphone may be fiddly but will be more accurate than typing it in the correct format. *Less strain on resources* Easier to understand interfaces means a user will complete tasks more quickly thereby freeing resources more quickly (servers etc.) *Less training and help needed* Users are less likely to rely on documentation or help centres, reducing running costs. Other answers possible.	**2**	1 mark each for any two relevant comments Answers must relate to user entry of topic and date using a smartphone or tablet.

Question			Expected Answer(s)	Max Mark	Additional Guidance
	(b)	(ii)	Correct description of sort – • Date/Issue 1& 2 compared, no swap • Date/Issue 3 &2 compared, swap • Date/Issue 2 &1 compared swap.	**2**	2 marks for all 3 correct; 1 mark for any 2 correct. Award max of 1 mark for correct description of insertion sort that doesn't refer to the scenario.
	(c)	(i)	One possible solution is provided below. Many possible alternatives exist. <FORM method="post" action="subscription.php" name="subscription" > Please enter a username (6 to 15 characters): <INPUT type="text" name="username" > Please enter a password (4 to 8 characters): <INPUT type="password" name="userpassword"> <INPUT type="submit" value="Submit Form" name="submit" > </FORM> <u>*Submit button alternative*</u> <BUTTON type=""submit">Submit Form</BUTTON>	**3**	1 mark for correct structure of <FORM> tag with method set as POST and action set to the correct file name. 1 mark for correct structure of <INPUT> tags used for username and password. Type attribute must be text for username; accept either text or password type for password. 1 mark for correct structure of <INPUT> tag for submit button; type must be correct; value must be correct. Note that type could be 'button' or 'submit'. Correct use of <BUTTON> tag as an alternative to this should receive 1 mark.

Question			Expected Answer(s)	Max Mark	Additional Guidance

		(ii)	One possible PHP solution is provided below. Many acceptable alternatives exist.
			
			$user=$_POST['username'];
			$userpassword=$_POST['userpassword'];
			
			$dbserver='magserver';
			$dbuser='subscribe';
			$dbpass='subpass';
			$dbname='subscribedata';
			$connection = mysql_connect($dbserver,$dbuser, $dbpass);
			mysql_select_db($database,$connection);
			
			$sql="INSERT INTO member VALUES ('$user', '$userpassword')";
			mysql_query($sql);
			
			mysql_close($connection);
			
			Many alternative solutions are possible.
				5	1 mark for assignment of both HTML values. (Step 1)
					
					1 mark for correct use of connection parameters;
					1 mark for connection to server and selection of database (mysql_connect and mysql_select_db code) (Step 2)
					
					1 mark for SQL query (Step 3)
					
					1 mark for close (Step 4)
					
					Notes
					• query and connection need not be assigned to variables
					• form data does not need to be validated
					• connection does not need to be checked

2.	(a)		Scope – what is to be included, in this case, for example, songs from 1990 to 1999 inclusive Constraint – the limits imposed, in this case, for example, only songs from top 40 singles chart.	**2**	1 mark for scope 1 mark for constraint Answers must relate to scenario.

Question			Expected Answer(s)	Max Mark	Additional Guidance
	(b)	(i)	Without permission from the creators or developers of the rival website, this would be a breach of their Intellectual Property Rights since IPR protects the visual design of their website. Copyright is an example of IPR. This gives the original creator exclusive rights to any artwork used on the website and use of those images by Radio Lowden would breach their copyright Trademarks are also examples of IPR. Since the rival radio station is likely to have used trademarked designs or logos on their website, use of these trademarks would be a breach of IPR.	**2**	1 mark each for any two relevant points. Many alternative answers are possible.
		(ii)	The test group would be given the test case and asked to carry out the task described. Since the test case indicated that the user cannot make use of a mouse, the test group would be required to perform the task using keyboard alternatives (or using speech input to control the movement of the screen pointer). The developers would observe the task being performed and make notes on any difficulties that the test group encountered. Since the test case requires alternatives to mouse input, difficulties with task completion would give the developers feedback regarding the accessibility of this page of the website.	**2**	1 mark for description of use made of test case by test group. This must acknowledge need for alternatives to mouse input. 1 mark for description of role of developers during observation of task performance. This must refer to difficulties encountered and accessibility of the page.

Question			Expected Answer(s)	Max Mark	Additional Guidance
	(c)	(i)	CREATE TABLE PlayList (ProgrammeID int NOT NULL, SongID text(6) NOT NULL, DatePlayed date NOT NULL, TimePlayed time NOT NULL, PRIMARY KEY (Dateplayed, Timeplayed), FOREIGN KEY (SongID) REFERENCES Song(SongID));	**4**	1 mark for correct structure of SQL CREATE statement 1 mark for four correct data types (char or varchar are acceptable alternatives to text for SongID) 1 mark for correct primary key Note: alternative primary key is possible (programmeID, songid, dateplayed) 1 mark for correct foreign key
		(ii)	SELECT Title FROM PlayList, Song WHERE PlayList.SongID = Song.SongID AND DatePlayed = '2016/05/26';	**2**	1 mark for SELECT with correct FROM 1 mark for correct WHERE clause with join
	(d)		1. IF front = back THEN 2. <queue is empty> 3. ELSE 4. <play playlist[front]> 5. SET front TO front-1 6. END IF	**3**	1 mark for queue empty check 1 mark for use of front pointer when removing from queue (step 4 or equivalent) 1 mark for update of front pointer
3.	(a)	(i)	*Difference object and class* Class is a template that defines the methods and attributes; object is an instance of a class. For example, teamName, anthem, flag and teamList are attributes within the class Team and an example of an object would be UK. *Encapsulation* Encapsulation is shown through the visibility on the (class) diagram. For example, access to the attributes, operations and class can be limited. In the diagram teamName is private and only available to the Team class; out with the Team class, it can only be accessed through use of the Team class methods. *Inheritance* Inheritance is demonstrated between classes Member and Athlete (or between Member and Official). The Athlete (or Official) class inherits all of the methods and attributes from the Member class.	**4**	2 marks for correct description of difference between class and object 1 mark for correct description of encapsulation 1 mark for correct description of inheritance Note that marks should only be awarded if descriptions make correct and appropriate reference to the class diagram.

Question			Expected Answer(s)	Max Mark	Additional Guidance
	(a)	(ii)	DECLARE team1 INITIALLY Team ("Brazil","Hino Nacional Brasileiro", "Bandeira do Brasil")	**1**	1 mark for correct use of team name, anthem and flag value Note that the variable declaration is not necessary to instantiate the object (DECLARE … INITIALLY is not necessary). Note also that answers that make use of OO languages such as Java and VB are acceptable alternatives to Haggis.
	(b)		DECLARE longjumpM AS ARRAY OF Athlete * 32	**2**	1 mark for declaration of array of Athlete objects 1 mark for correct dimension
	(c)	(i)	Hello, my name is Ali I'm an athlete on the team. Hello, my name is Omar. Hello, my name is Nour.	**1**	1 mark for correct output produced by calls
		(ii)	The procedure call invokes the introduce method of the myTeam object. teamList is an array of Members which can store objects of type Member, Athlete or Official. Since Athlete inherits methods and properties from Member, the introduce method in the Athlete class overrides the introduce method in the super class: since the member with firstName Ali is an athlete, the output is produced by applying the overriding method. Official also inherits methods and properties from Member but since it has no overriding method of its own, the output for the member with firstName Nour is produced by applying the method in the superclass. The output for the object of class Member with the firstName Omar is straightforwardly generated by the introduce method in Member.	**2**	1 mark each for any two relevant points that make appropriate use of relevant OO terminology Examples of OO terminology that could be discussed in the answers include: • (super) class and sub-classes • inheritance • polymorphism

Question			Expected Answer(s)	Max Mark	Additional Guidance
	(d)	(i)	Pop Australia, pop France, push USA, push France, push Australia	**2**	1 mark for correct pop operations 1 mark for correct push operations
		(ii)	A new node would be added to store Russia and its pointer would be set to 241 (team Germany). The pointer of the node containing USA would be updated; instead of storing 241, it would now store the address of the new node.	**2**	1 mark for addition of new node containing Russia and correct value of its pointer 1 mark for updating pointer of node containing USA with address of the new node
	(e)		Analytics will provide insights and patterns that aid decision making. For example, analysis of ticketing and competition data could benefit the Management Committee by providing them with details about which events were popular with the general public from across the world and which areas of the arenas sold quickly. Analytics could show whether event popularity varied from country to country and whether it was dependent on the participation of particular athletes. This could help the Committee to determine their pricing and distribution of tickets for the Games in 2020.	**1**	1 mark for description of one benefit of analytics for the Management Committee Description must make reference to one of the sources of big data listed in the question and must illustrate how analytics could be applied to that data to benefit the Committee.
4.	(a)	(i)	1. RECORD pupil IS { INTEGER PupilID, STRING FirstName, STRING LastName, STRING DateOfBirth, STRING RegClass } 2. DECLARE details AS ARRAY OF pupil * 10 INITIALLY [ ] 3. importData (details[ ]) 4. search(details[ ])	**3**	1 mark for definition of record structure with 5 fields and correct field types 1 mark for declaration of array of records with 10 records 1 mark for both procedure calls

Question			Expected Answer(s)	Max Mark	Additional Guidance
	(a)	(ii)	1. SET lower TO 0 2. SET upper TO 9 3. SET found TO false 4. REPEAT 5. SET mid TO (lower + upper)/2 6. IF pupilData[mid].PupilID = 112213 THEN SET found TO true 7. IF pupilData[mid].PupilID < 112213 THEN SET lower TO mid + 1 8. IF pupilData[mid].PupilID > 112213 THEN SET upper TO mid - 1 9. UNTIL upper < lower OR found = true 10. IF found = true THEN 11. SEND pupilData[mid].pupilID, pupilData[mid].FirstName, pupilData[mid].LastName, pupilData[mid].DateOfBirth, pupilData[mid].RegClass TO DISPLAY 12. ELSE 13. SEND "Pupil ID not found" TO DISPLAY 14. END IF	**5**	1 mark for initialisation 1 mark for conditional loop with correct complex condition 1 mark for use of array of records with correct indexing 1 mark for comparisons and appropriate updates 1 mark for display of correct pupil details (note that the check for found is not essential).
	(b)		1. < *connect to database server* > 3. SET query TO "update pupildata set RegClass = '6B' where PupilID = 112213" 4. < *execute SQL query* > 5. < *close database connection* >	**3**	1 mark for connect / close connection with database server 1 mark for correct record (where clause) 1 mark for update query
	(c)		*Line 1* # Sort Algorithm Used: Bubble Sort *Line 5* IF pupilData[counter-1] > pupilData[counter] THEN	**2**	1 mark for Line 1 correct 1 mark for Line 5 correct Alternative / equivalent forms of line 5 are acceptable.

Question			Expected Answer(s)	Max Mark	Additional Guidance
	(d)		Quicksort takes less time than bubble sort to sort large data sets since fewer comparisons are needed. Based on average case scenario, quicksort would need 21932 comparisons (ie $n\log_2(n)$ ie $2000 \times \log_2(2000) = 21,932$) whereas bubble sort will need n^2 (ie $2000^2 = 4,000,000$).	**2**	1 mark for efficiency of quicksort. Note that 'quicker' without further detail is not sufficient. 1 mark for justification of this efficiency. Note that justification in terms of Big O notation is not essential.

[END OF MARKING INSTRUCTIONS]