



National
Qualifications

CS(H)24AMS

Computing Science
Marking Instructions

perfectpapers

© 2023-2024 Perfect Papers – All rights reserved

General Marking Principles for Higher Computing Science

Always apply these general principles. Use them in conjunction with the detailed marking instructions, which identify the key features required in candidates' responses.

- (a) Always use positive marking. This means candidates accumulate marks for the demonstration of relevant skills, knowledge and understanding; marks are not deducted for errors or omissions.
- (b) If a 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.
- (c) Award marks regardless of spelling, as long as the meaning is unambiguous
- (d) Candidates may answer programming questions in any appropriate programming language or pseudocode. Award marks regardless of minor syntax errors, as long as the intention of the coding is clear.
- (e) For a describe question, candidates must provide a statement or structure of characteristics and/or features. This will be more than an outline or a list. It may refer to, for example, a concept, process, experiment, situation, or facts, in the context of and appropriate to the question. Candidates must make the same number of factual/appropriate points as there are marks available in the question.
- (f) For an explain question, candidates must relate cause and effect and/or make relationships between things clear, in the context of the question or a specific area within the question.

Number			Question	Instructions	Marks
1			Convert the 8-bit two's complement number 1101 1101 into denary.	-35 (1 mark)	1
2			A string variable called <code>warehouse</code> has been assigned the value 'Dalkeith' and another string variable called <code>postcode</code> has been assigned the value 'EH23 8TQ' as shown below. (see paper) The variable <code>warehouseCode</code> is to be assigned the value 'Dal8TQ' using substring operations. Using a programming language of your choice write line 3.	SET <code>warehouseCode</code> TO <code>left(warehouse,3) & right(postcode,3)</code> <code>warehouseCode = warehouse[1:3] & postcode[6:8]</code> 1 mark for both substrings 1 mark for assignment and concatenation	2
3			Explain why a processor with wider data bus would outperform an identical processor with a smaller data bus.	The width of the database determines how much data can be moved between processor/main memory in one operation (1 mark) The wider the data bus the more data can be transferred in one operation, which leads to an increase in performance. (1 mark)	2
4	a		Convert the binary number below into floating point representation. -0.0001 0111 0010 1100 There are 16 bits for the mantissa (including the sign bit) and 8 bits for the exponent.	• Sign: 0 • Mantissa: 101 1100 1011 0000 • Exponent: 0000 1011 1 mark each bullet. Max 3 marks	3
5	a		Complete the missing steps of the fetch-execute cycle.	• The address bus is set up with the address (by the processor) • The processor activates the read line on the control bus. 1 mark each bullet, max 2 marks	2

Number			Question	Instructions	Marks
	b		Other than the data bus width, several factors can improve the performance of the fetch-execute cycle. State one of these factors and explain why it improves performance.	Factor: • Clock Speed • Cache • Number of processor cores 1 mark for any bullet Explanation: Affects the number of instructions executed 1 mark for any bullet	2
6			Describe the role of the client when developing software using agile methodologies.	The client: • Details requirements • Outlines scope and boundaries • Evaluates prototype/suggests changes • Provides feedback/working with development team throughout process 1 mark each bullet, max 2 marks	2
7			A function has been written to generate numbers until a limit value is reached. The function receives three parameters and calculates an array of integers by adding the previous two numbers to create the next one. Numbers are generated while they are less than the limit given. For example, if the parameters are first = 3, second = 6 and limit = 31. Then the numbers generated will be: 3, 6, 9, 15, 24 All numbers in the sequence must be smaller than the limit. The function is shown below. (see paper)		

Number		Question	Instructions	Marks								
	a	A logic error in the code means an incorrect array of numbers if generated. The trace table below shows the line numbers where a variable has changed. (see paper) State the missing values at A, B and C.	A = [3, 6, 9, 15, 0, 0, 0] B = [3, 6, 9, 15, 24, 0, 0] C = [3, 6, 9, 15, 24, 39, 0] 1 mark each, max 3 marks	3								
	b	Re-write Line 21 of the code to generate the correct sequence.	WHILE (seqArray[index-1] + seqArray[index-2])< limit DO 1 mark for using seqArray[index-1] + seqArray[index-2] 1 mark for bracketing condition	2								
	c	Identify one formal parameter and its associated actual parameter.	<table><tr><th>Formal</th><th>Actual</th></tr><tr><td>first</td><td>1stVal</td></tr><tr><td>second</td><td>2ndVal</td></tr><tr><td>limit</td><td>myLimit</td></tr></table> Award 2 marks for correct pair of parameters. Award 1 mark for any correct formal/actual parameter. Award 1 mark for correct pair of parameters wrong way round.	Formal	Actual	first	1stVal	second	2ndVal	limit	myLimit	2
Formal	Actual											
first	1stVal											
second	2ndVal											
limit	myLimit											
	d	Describe the operation of Line 89 during the execution of this program.	• Calls the function using the given (actual) parameters. • Assigns the returned array to the variable mySequence. 1 mark each bullet, max 2 marks	2								
	e	Programmers have control over the scope of a variable when writing code. Describe how the position of the declaration of a variable, within code, determines its scope.	Declaration inside a function/procedure (1 mark) then only accessible within that function/procedure (therefore local) (1 mark). OR Declaration outside subprograms subprograms (1 mark). then accessible throughout entire program (therefore global) (1 mark).	2								

Number			Question	Instructions	Marks
8			A car dealership sells a range of 80 different cars. Sample data about the cars is shown below. (see paper) A program is designed to track car sales.		
	a	i	Using a programming language of your choice, define a suitable record data structure for the car data above. The record data structure should be called <code>car</code>.	<pre>RECORD car IS (STRING Make STRING Model STRING EngineSize STRING FuelType INTEGER Price INTEGER NumberInStock)</pre> - Record structure with name <code>car</code> - Six fields with appropriate data types 1 mark each bullet, max 2 marks	2
		ii	Using a programming language of your choice, declare a variable called <code>carstock</code> which can store the details of 150 cars. Your answer should use the record data structure created in part (i).	<pre>DECLARE carstock AS ARRAY OF car INITIALLY [] * 150</pre> VB.net <pre>Dim carstock() As Car = New Car(150) {}</pre> Java <pre>Car[] carstock = new Car[150];</pre> Python <pre>carstock = [Car() for i in range(150)]</pre> - Array structure and size (using name <code>carstock</code>) - Using <code>car</code> data type 1 mark each bullet, max 2 marks	2

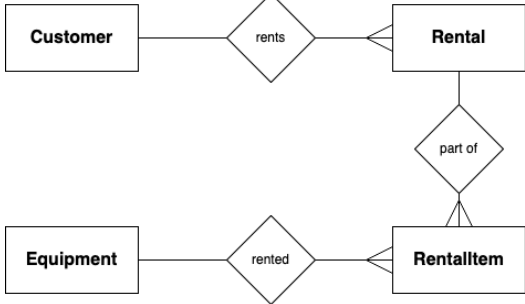
Number		Question	Instructions	Marks
	b	Customers often have questions of the sales people at the car dealership. For example: ‘Which is the cheapest electric car that you have that is under £20,000 and is in stock?’ A car is in stock if the stock level is greater than zero. The top-level design for the algorithm to answer this question is shown diagram below. Complete the diagram below to show the missing data flows for the last two modules.	1 mark for inputs to module 3 1 mark for outputs from module 3 1 mark for inputs to module 4 <i>Accept alternative solution</i>	
		Solution 1 <pre> graph TD CS[Car Sales] --> L[Load data structure with data from text file] CS --> A[Ask users to enter maximum price and fuel type.] CS --> S[Search for the cheapest car that meets the requirements if one exists.] CS --> D[Display the make, model and price of the car or a message saying that no car has been found.] L --> carstock() CS A --> maxPrice, fuelType CS S --> maxPrice, fuelType, carstock() CS S --> found, position CS D --> position, carstock(), found CS </pre>		
		Solution 2 <pre> graph TD CS[Car Sales] --> L[Load data structure with data from text file] CS --> A[Ask users to enter maximum price, car type and stock requirement.] CS --> S[Search for the cheapest car that meets the requirements if one exists.] CS --> D[Display the make, model and price of the car or a message saying that no car has been found.] L --> carstock() CS A --> maxPrice, fuelType CS S --> maxPrice, fuelType, carstock() CS S --> found, foundMake, foundModel, foundPrice CS D --> found, foundMake, foundModel, foundPrice CS </pre>		

Number			Question	Instructions	Marks
	c		The third module of the program searches for the cheapest car that meets the requirements. A variable <code>cheapestPrice</code> is initially set to the price of the most expensive car.		
		i	Using a design technique of your choice, design a solution to initially set the variable <code>cheapestPrice</code>.	- Initial <code>cheapestPrice</code> to value of first record - Loop for remaining/all cars - IF <code>carstock[index].Price > cheapestPrice</code> - Updating <code>cheapestPrice</code> 1 mark each, maximum 4 marks	4
			FINDING MAX <pre> SET cheapestPrice TO carstock[0].Price FOR index = 1 TO 149 DO IF carstock[index].Price > cheapestPrice THEN SET cheapestPrice TO carstock[index].Price ENDIF END FOR </pre>		
		ii	Using a programming language of your choice, write the code for the remaining part of module 3. (see paper) If no car is found then a variable called <code>found</code> is set to <code>false</code>. Your answer should use the data structure created in part(a). Assume that <code>cheapestPrice</code> has initially been set to the price of the most expensive car.	- use of array of records - with matching fields from a(i) <code>carstock[index].Price < maxPrice AND</code> <code>carstock[index].FuelType = fuelType</code> <code>carstock[index].Price < cheapestPrice</code> <code>found</code> initialised to <code>false</code>, set to <code>true</code> if car found 1 mark each bullet, max 6 marks	6

Number			Question	Instructions	Marks
			<pre> SET position TO -1 SET found TO FALSE FOR index FROM 0 TO 149 DO IF carstock[index].Price < maxPrice AND carstock[index].FuelType = fuelType AND carstock[index].Price < cheapestPrice THEN SET cheapestPrice TO carstock[index].Price SET position TO index SET found TO true END IF </pre>		
9			A game displays several letters and then asks the player to create words using some or all of the letters. For each set of letters displayed, a target number of words is requested. When a correct word is entered, the word is displayed on screen. Guessing the word again, displays an error message. As each correct word is entered, the target number of words is reduced. When the number of words requested has been reached, the next set of letters is displayed with a new target. (see paper)		
	a		Describe two process of this game.	• Check word is valid • Check word hasn't been guessed already • Reduce target when correct guess • Other valid 1 mark for any one bullet	2

Number			Question	Instructions	Marks
	b		Each set of letters is read from a file. The files are named according to the level e.g. <code>level1.txt</code>, <code>level2.txt</code>, <code>level3.txt</code> etc. The first entry in each file contains the letters to be displayed to the player for the level. The remaining entries contain all the valid words for the combination of letter e.g. Text file: <code>level3.txt</code> File Contents ATM MAT TAM AM AT MA TA Using a design technique of your choice, design a program to read in this file, store the letters to be shown to the player and the valid words.	• Open text file for reading using the level number in the filename • set letters to first entry in file • Read remainder of file into data structure • Loop for reading from file 1 mark for each bullet	4
			SET file TO "level" & levelnumber & ".txt" OPEN file DECLARE letters AS STRING INITIALLY FROM sqaData DECLARE words AS ARRAY OF STRING SET index TO 0 WHILE <not end of file> DO RECEIVE data FROM file SET words[index] TO data SET index TO index + 1 END WHILE		
	c		Part of the quiz program is shown below. (see paper) Describe how lines 45 to 57 could be amended to make the code more efficient.	• Change the fixed loop (FOR) to a conditional loop (WHILE) • WHILE valid=false AND index < LENGTH(words) • Add an index variable which is incremented on each step 1 mark for each bullet, max 2 marks	2

Number			Question	Instructions	Marks
	d		Describe how the arrays <code>words</code> and <code>guessed</code> are used in this code to ensure that a valid word can only be guessed once.	• Concept of parallel arrays • Binary value of <code>guessed</code> associated with word in the same index position • <code>guessed[index]</code> set to true when <code>word[index]</code> is guessed. 1 mark for association / use of parallel arrays 1 mark for setting guessed when word entered.	2
	e		The program is implemented using modular code. Describe one benefit of implementing modular code.	• Facilitates module libraries which allows the reuse of code • Each part of the solution can be tested independently. • Errors are easier to locate. • Different programmers can code different modules (independently). • Variables only in memory during execution of subroutine. • Limits accessibility of local variables. 1 mark for any one bullet	1
	f		The program uses Unicode for the level text files. Describe an advantage of using Unicode over ASCII, referring to the number of bits used to represent a character in each format.	Unicode can represent more characters. (1 mark) 1 mark for any one bullet from 216 when compared to ASCII $2^8/2^7$. OR uses 16 bits compared to 8/7. Allow alternatives that reference UTF - 8, UTF -16, UTF -32 or UCS -2 (17×2^{16}) <i>More languages on its own does not get first mark</i>	2

Number		Question	Instructions	Marks
10		A snowboard rental shop uses a relational database to store information about the equipment they have for rent, the customers who rent equipment, the rentals that are made, and the rental items that are associated with each rental. The four tables in the database are as follows: (see paper) Draw an entity-relationship diagram to show the relationships that exist in this database. Your answer should show the entity names and cardinality. Attributes are not required on the diagram.	 • Customer 1:M Rental • Rental 1:M RentalItem • Equipment 1:M RentalItem 1 mark each bullet, max 3 marks	3
11		An ice hockey league uses a database to store details of players and their stats. A sample of the data stored in the Player table is shown below. (see paper) The national coach would like to produce a report displaying the lowest and highest number of <code>Assists</code> for each team with “Dundee” in their name as shown below. (see paper) When tested, the actual output did not match the expected output. Identify the three errors in the above SQL statement.	• Max and Min functions are in incorrect positions. They should be swapped. • The Where clause wildcard “Dundee*” will only find teams where the name begins with Dundee. It should be “*Dundee”. • There is no GROUP BY clause, it should be GROUP BY Team. 1 mark each bullet, max 3 marks	3
12		A tour company uses a relational database to store the following information in three tables as shown below. (see paper)		

Number			Question	Instructions		Marks					
	a		When the Booking table was originally designed, it was suggested that a compound key could have been used. Explain why a compound key would not have been suitable for the Booking table.	• bookingID is able to uniquely identify the row • The primary keys from the other tables would not uniquely identify a row. 1 mark each bullet, max 2 marks		2					
	b		Some data from the booking table is shown below. (see paper) Complete the table below showing the expected output from the following SQL statement. <pre>SELECT tourDate, SUM(reserved) as 'Number in Group' FROM booking WHERE tourID = 1 GROUP BY tourDate</pre>	<table><tr><th>tourDate</th><th>Number In Group</th></tr><tr><td>2023-10-28</td><td>11</td></tr><tr><td>2023-10-30</td><td>8</td></tr></table> • Grouping on tourDate to return two rows • Correct sum of grouped values 1 mark each bullet, max 2 marks	tourDate	Number In Group	2023-10-28	11	2023-10-30	8	2
tourDate	Number In Group										
2023-10-28	11										
2023-10-30	8										
	c		The database is improved by adding a table for guides. (see paper) The design of the booking table is changed as a result. Complete the data dictionary below for the changes to the booking table.	Attribute: guideID (replaces guideName) Key: Foreign Key / FK Type: Integer Validation: Restricted choice (loop up from guide) All complete 2 marks, 1 error 1 mark, Otherwise 0		2					

Number			Question	Instructions		Marks
13			A company sells a variety of products, such as clothes, electronics, and home goods. The company offers a warranty on all its products, ranging from 1 year to 5 years in duration. The company wants to track the warranty status of each product, including the purchase date, the warranty period in years, and when it expires. The company also wants to track the customers who purchased each product. The company uses a database with four tables for this purpose. The four tables used are shown below. (see paper)			
	a		Design a query to display the number of warranties that the customer with <code>custID</code> of 10 has that expire in 2024.	Field(s) and Calculation(s)	COUNT(*)	2
				Table(s)	warranty, sale	
				Search Criteria	custID=10 AND expires >=01/01/2024 AND expires <=31/12/2024	
				Grouping		
				- COUNT with both tables - WHERE condition 1 mark each bullet, max 2 marks		
	b		A product with <code>productID</code> 3, sold with a <code>sale_date</code> before 01/10/2023 is to be recalled because it is unsafe. A list of all customers who purchased the product is required. Complete the design of a query to display the full name and mobile number of every customer affected.	Field(s) and Calculation(s)	firstname, lastname, mobile	2
				Table(s)	customer, sale	
				Search Criteria	sale.productID =3 AND sale.sale_date < 01/10/2023	
				Grouping	(custID)	
				Accept alternatives that use <i>productID</i> in <i>product</i> - Three required fields - Two criteria with AND 1 mark each bullet, max 2 marks		

Number			Question	Instructions	Marks
	c		Customers who purchased any Smart TV with a 5 year warranty are to have an extra year added to their warranty. Write the SQL statement that will implement this.	- Update with three tables - SET years = years + 1 - warranty.years > 5 AND product.description LIKE "*Smart TV*" - Joins across the three tables 1 mark each bullet, max 4 marks	4
			<pre>UPDATE warranty, sale, product SET years = years + 1 WHERE warranty.years >= 5 AND product.description LIKE "*Smart TV*" AND product.productID = sale.productID AND sale.saleID = warranty.saleID</pre>		
	d		The company want to calculate the average sale_price.		
		i	Write the SQL statement to display the average sale price as shown below. (see paper)	SELECT avg(sale_price) AS "Average Sale Price" FROM sale - Aggregate function AVG on sale_price - alias and table 1 mark each bullet, max 2 marks	2
		ii	The query from part (i) is saved as 'AvgSalePrice'. Using this query, complete the SQL statement to display customers who have spent more than the average sale price, in any sale, from highest to lowest as shown below. (see paper)	- FROM statement with three tables and saved query - Equi-join to link three tables and saved query included - Condition with reference to saved query - Sort by sale_price descending 1 mark each bullet, max 4 marks	4
			<pre>SELECT firstname, lastname, sale_price, product_name FROM sale, product, customer, AvgSalePrice WHERE sale.custID = customer.custID AND product.productID = sale.productID AND sale_price > 'Average Sale Price' ORDER BY sale_price DESC</pre>		

Number			Question	Instructions	Marks
14			HTML 5 elements have been used to define the parts of a web page which is shown below. (see paper) State which elements should be used for the parts labelled A and B.	A - Nav B - Footer 1 mark each, max 2 marks	2
15			A digital “escape room” contains challenges which are completed by participants to gain a code which allows them to leave the room and gain a prize. One of the codes is displayed when a hidden area of a web page is clicked. When the participant moves the mouse over the clickable area of the web page, text appears which says “Click me”. The text disappears when the mouse moves off the hidden area. Clicking displays a code. An example of the intended effect is shown below. (see paper) The code used to implement this feature is shown below. (see paper)		
	a		State two reasons why this code is not fit for purpose.	• Button element has no onClick event so does not show the message. • Button has no onMouseOut event to hide the message. 1 mark each bullet, max 2 marks	2
	b		Explain why the area changes colour when the mouse moves over it.	The CSS for the #button id makes use of the hover property (1 mark) When the mouse hovers over the button, the CSS rule changes the background colour to blue (1 mark).	2

Number			Question	Instructions	Marks
16			A new web site is being developed. This site will allow users to upload and share documents, program code, images and other types of files. Once uploaded, these files can be used in collaborative projects that can be shared with other users. Within each project, users can discuss and exchange ideas with others and track the progress of the project. Users should sign into the site using a username and password. Once signed in, users should be able to access one of four pages - an explore page, a profile page, a “my projects” page and a settings page. From the explore page, users can view a page showing projects created by others and a join page that allows them to join project. On the “my projects” page, users can view the projects they have joined and they can view a page with “recent updates” from their projects and another page where they can upload files to projects. From the profile page they can access a page to edit their profile.		
	a		State one functional requirement of this website.	Use of forms for: • add/read updates • user sign-in • edit profile/ join projects An element to add files / access files Any of the above for 1 mark	1
	b		Draw the navigational structure of this website.	1 mark for level 1 complete 1 mark for level 2 complete Max 2 marks	2

Number		Question	Instructions	Marks
		Home/Sign-in explore profile my projects settings view projects join project edit profile view joined projects recent updates upload files Level 1 Level 2		
	c	The developers proposed an initial design for the “my projects” page. The design is shown below. (see paper) A low fidelity prototype was created from the wireframe above. State two issues that should be identified from usability testing.	• No navigation for across the site • Blank button which has no function next to Recent Updates • No site header 1 mark each bullet, max 2 marks	2
	d	A proposed design for the recent updates page is shown below. (see paper)		

Number			Question	Instructions	Marks
		i	The style rule for h1 headings is shown below. <pre>h1 { font-family: Verdana; text-align: left;}</pre> All h2 and h3 elements should also be formatted using the rule above. All h2 elements should also have a 15 pixel padding. Making use of a grouping selector where appropriate, write the CSS rules to format the headings.	- h1, h2, h3 grouping selector - h2 with 15px padding rule 1 mark each bullet, max 2 marks <pre>h1,h2,h3 { font-family: Verdana; text-align: left; }</pre> <pre>h2 { padding: 15; }</pre> or <pre>h2 { padding:15 15 15 15; }</pre> <i>15px optional in all cases.</i>	2
			The navigation bar shown makes use of the following CSS. (see paper)		
		ii	Explain why descendant selectors are used here.	- style will only apply within the (parent) nav element - style will apply to unordered list/list and/or anchor elements 1 mark each bullet, max 2 marks	2
		iii	The hyperlinks work if the user clicks on or near the text. Identify the line of code that allows this to happen.	display:block (1 mark)	1

Number			Question	Instructions	Marks
17			A web site is used by users to find jobs in their local community. Users complete a search profile which matches them to jobs on the site. The search profile page allows users to enter the following details. • firstname • lastname • email address • phone number • type of work - part-time, full-time, volunteering • interested in - retail, hospitality, customer service, other		
	a		Using the information above, draw a wireframe design for this web page.	• labelled fields for <i>firstname</i>, <i>lastname</i>, <i>email</i>, <i>phone number</i> • Validated selection for <i>type of work</i> and <i>interested in</i> • Submit button 1 mark each bullet, max 3 marks	3
	b		Part of a wireframe for the search results page of the web site is shown below. (see paper)		
		i	Part of the HTML for this page is shown below. (see paper) Write the CSS rule for <code>.job-listing p</code> so that it is efficiently formatted.	<pre>.job-listing p { font-size: 14px; margin: 0px; }</pre> 1 mark for margin/padding together 1 mark for font-size <i>Only 1 mark if includes width: 500px and/or padding: 10px as these would be set in .job-list</i>	2

Number			Question	Instructions	Marks
		ii	Write a CSS rule to display the company logos to match the design wireframe.	<pre>.logo { width: 100px; height: 100px; float: right; margin: 10px; }</pre> 1 mark for dimensions and margin with .logo rule 1 mark for use of float	2
	c		Once the website has been completed, compatibility testing is carried out. During compatibility testing several comments were made. (see paper) Give two reasons why users may have different experiences when testing the same website.	• Different device/screen sizes/resolution would make layouts different • Different browsers (some colours styling would look different) 1 mark each bullet, max 2 marks	2

Detail of Sources / Mark Allocations and Balance

Qu	Section	Type	Skills	Content	Marks	QuType	CAS Item	A type justification
1	1	short answer	Application of knowledge of computer systems	Computer systems (CS)	1	C	Conversion of two's complement numbers from binary to denary and vice versa.	
2	1	short answer	Implementation (read and write code)	Software Design and Development (SDD)	2	C	Substring operations	
3	1	short answer	Application of knowledge of computer systems	Computer systems (CS)	2	C	Width of data bus	
4	1	short answer	Application of knowledge of computer systems	Computer systems (CS)	3	C	Floating point representation	
5a	1	short answer	Application of knowledge of computer systems	Computer systems (CS)	2	C	Fetch execute cycle	
5b	1	short answer	Application of knowledge of computer systems	Computer systems (CS)	2	C	Fetch execute cycle	
6	1	short answer	Analysis	Software Design and Development (SDD)	2	C	Agile methodologies	

Qu	Section	Type	Skills	Content	Marks	QuType	CAS Item	A type justification
7a	1	extended answer	Testing	Software Design and Development (SDD)	3	A	dry runs / trace tables	identifying errors in code and their effect and in particular, the use of trace tables after initialisation of variables or the first iteration of a loop, especially where a condition is involved
7b	1	extended answer	Testing	Software Design and Development (SDD)	2	A	Identification of logic errors at this level	complex conditions
7c	1	extended answer	Implementation (read and write code)	Software Design and Development (SDD)	2	A	Parameter passing (formal and actual)	correct use of parameter passing (formal and actual)
7d	1	extended answer	Implementation (read and write code)	Software Design and Development (SDD)	2	C	sub-programs / functions - read and explain code.	
7e	1	extended answer	Implementation (read and write code)	Software Design and Development (SDD)	2	C	the scope of local and global variables	
8ai	1	extended answer	Implementation (read and write code)	Software Design and Development (SDD)	2	C	Record structure	
8aii	1	extended answer	Implementation (read and write code)	Software Design and Development (SDD)	2	C	Array of records	

Qu	Section	Type	Skills	Content	Marks	QuType	CAS Item	A type justification
8b	1	extended answer	Design	Software Design and Development (SDD)	3	A	Data flow	data flow when designing solutions
8ci	1	extended answer	Design	Software Design and Development (SDD)	4	A	Finding Maximum	array of records
8cii	1	extended answer	Implementation (read and write code)	Software Design and Development (SDD)	6	A	Array of records	complex conditions
9a	1	extended answer	Analysis	Software Design and Development (SDD)	2	C	Input/Process/Output	
9b	1	extended answer	Design	Software Design and Development (SDD)	4	C	Implement efficient design solutions	
9c	1	extended answer	Evaluation	Software Design and Development (SDD)	2	A	Efficient use of coding constructs	problem solving to improve the efficiency of code
9d	1	extended answer	Implementation (read and write code)	Software Design and Development (SDD)	1	C	Parallel Arrays	
9e	1	extended answer	Evaluation	Software Design and Development (SDD)	2	C	Advantages of modular code	
9f	1	extended answer	Application of knowledge of computer systems	Computer systems (CS)	2	C	Unicode	

Qu	Section	Type	Skills	Content	Marks	QuType	CAS Item	A type justification
10	2	short answer	Design	Database Design and Development (DDD)	3	C	Entity-relationship diagrams	
11	2	short answer	Implementation (read and write code)	Database Design and Development (DDD)	3	C	Implementation (read and write code)	
12a	2	extended answer	Analysis	Database Design and Development (DDD)	1	C	Compound key	
12b	2	extended answer	Testing	Database Design and Development (DDD)	2	C	Expected output from SQL	
12c	2	extended answer	Design	Database Design and Development (DDD)	2	C	Data dictionary / validation	
13a	2	extended answer	Design	Database Design and Development (DDD)	2	C	Design a solution to a query	
13b	2	extended answer	Design	Database Design and Development (DDD)	2	C	Design a solution to a query	
13c	2	extended answer	Implementation (read and write code)	Database Design and Development (DDD)	4	A	Use of SQL operations	any problem that requires data from multiple tables requiring the use of join(s)

Qu	Section	Type	Skills	Content	Marks	QuType	CAS Item	A type justification
13di	2	extended answer	Implementation (read and write code)	Database Design and Development (DDD)	2	C	Use of SQL operations	
13dii	2	extended answer	Implementation (read and write code)	Database Design and Development (DDD)	4	A	Use of SQL operations	any problem that requires data from multiple tables requiring the use of join(s)
14	3	short answer	Implementation (read and write code)	Web Design and Development (WDD)	2	C	HTML code / Implementation HTML	
15a	3	short answer	Implementation (read and write code)	Web Design and Development (WDD)	2	C	JavaScript - onClick / onMouseOut	
15b	3	short answer	Implementation (read and write code)	Web Design and Development (WDD)	2	C	CSS - hover	
16a	3	extended answer	Analysis	Web Design and Development (WDD)	1	C	Functional Requirements	
16b	3	extended answer	Design	Web Design and Development (WDD)	2	C	web site structure	
16c	3	extended answer	Testing	Web Design and Development (WDD)	2	C	Usability testing	

Qu	Section	Type	Skills	Content	Marks	QuType	CAS Item	A type justification
16di	3	extended answer	Implementation (read and write code)	Web Design and Development (WDD)	2	A	CSS grouping selectors	use of grouping and descendant selectors
16dii	3	extended answer	Implementation (read and write code)	Web Design and Development (WDD)	2	A	CSS descendant selectors	use of grouping and descendant selectors
16diii	3	extended answer	Implementation (read and write code)	Web Design and Development (WDD)	1	C	CSS: display: block	
17a	3	extended answer	Design	Web Design and Development (WDD)	3	C	Wireframe design	
17bi	3	extended answer	Implementation (read and write code)	Web Design and Development (WDD)	2	A	CSS: grouping / descendant selectors	using CSS to control appearance and positioning to match a wireframe design
17bii	3	extended answer	Implementation (read and write code)	Web Design and Development (WDD)	2	A	CSS: float	using CSS to control appearance and positioning to match a wireframe design
17c	3	extended answer	Testing	Web Design and Development (WDD)	2	C	Compatibility	

	Paper	Section 1 (Actual)	Section 2 (Actual)	Section 3 (Actual)
Short Response Questions	18	14	6	6
Extended Reponse Questions	62	41	19	19
	Totals	55	25	25

Proportion of A and C questions	Section 1	Section 2	Section 3	Section 1 + 2	Section 1 + 3
A Marks	22	8	8	30	30
C Marks	33	17	17	50	50
			Total	80	80

		Analysis	Design	Implementation (read and write code)	Testing	Evaluation	Application of knowledge of computer systems	Totals
Software Design and Development (SDD)	Approx. 40	4	11	20	5	3	0	44
Database Design and Development (DDD)	Approx. 25	1	9	13	2	0	0	25
Web Design and Development (WDD)	Approx. 25	1	5	15	4	0	0	25
Computer systems (CS)	Approx. 15	0	0	0	0	0	12	12
Notional		5%	30%	40%	10%	5%	10%	
Actual		6%	24%	46%	10%	3%	11%	