



National
Qualifications

CS(AH)21

Computing Science
Marking Instructions

perfectpapers

© 2020-2021 Perfect Papers – All rights reserved

| Number | | Question | Instructions | Marks |
|--------|---|---|--|-------|
| 1 | a | Describe the use of a constructor in object-oriented programming. | The constructor assigns initial values upon instantiation of an object (1 mark) <i>Must refer to instantiation/creation of an object for mark</i> | 1 |
| | b | State the feature of object-oriented languages that requires that methods are used to access instance variables rather than directly accessing the instance variables themselves. | Encapsulation (1 mark) | 1 |
| 2 | | A web site is available for a range of devices. It uses the following CSS code. (see paper) Explain the purpose of this code. | This code changes the background colour for different screen sizes (1 mark) of screen. There are three display options (1 mark) | 2 |
| 3 | | The database is held on a server with the following access details. (see paper) | | |
| | a | Write the PHP code to connect to this database. | <code>\$conn = mysqli_connect("db.myrental.com","dbsecure","cloud812","myrentaldb");</code> 1 mark for mysqli_connect 1 mark for all given connection values | 2 |
| | b | Write the PHP code to terminate the connection to this database. | <code>mysqli_close();</code> 1 mark | 1 |
| 4 | | Some sample data for a table from a database is shown below. (see paper) The primary key is ClownNo and LocationID is a foreign key from a table called Location. Write the SQL statement to create the Clown table. | 1 mark for CREATE TABLE with name 1 mark for attributes, types 1 mark for Primary Key 1 mark for Foreign Key <code>CREATE TABLE Clown (ClownNo VARCHAR(6) PRIMARY KEY, ClownType VARCHAR(10) NOT NULL, LocationID NOT NULL, FOREIGN KEY (LocationID) REFERENCES Location(LocationID))</code> | 4 |
| 5. | | During the development of the game, the developers make use of integrative testing. Describe the purpose of integrative testing. | Integration testing takes modules that have been unit tested, groups them in larger sets of code (1 mark) and tests the integration of the modules functions as expected (1 mark) . | 2 |

| Number | | Question | Instructions | Marks |
|--------|---|--|---|-------|
| 6. | | When customers access the online service web site a PHP session begins. | | |
| | a | Describe the purpose of the PHP session. | A session is used to store information in variables to be used across multiple pages. 1 mark | 1 |
| | b | State the command used to begin the session. | session_start() 1 mark | 1 |
| 7 | | A book shop has developed an app, using an object-oriented programming language, to track details of books that they stock. A simplified version of the UML class diagram and code for the Book class is shown below. (see paper) | | |
| | a | Using a programming language of your choice, write the code equivalent to Line 1 of the Book class, for the declaration of the eBook class. | CLASS eBook INHERITS Book WITH { STRING DownloadLink, INTEGER QtySold } 1 mark for class declaration indicating use of inheritance 1 mark for attributes (1 mark for 5 attributes if no inheritance) | 2 |
| | b | Information about a physical book to be added to the app is shown below. (see paper) Using a programming language of your choice: | | |
| | i | Write code to create an array of authors using the values provided. | DECLARE author1 INITIALLY Author ("Chiyo Abad", "English") DECLARE author2 INITIALLY Author ("Sai Rashi", "Urdu") DECLARE myAuthors AS ARRAY OF Author INITIALLY [author1, author2] 1 mark for instantiating each author (2 marks available) 1 mark for assigning these to an array of Author <i>Range of responses possible</i> | 3 |

| Number | | Question | Instructions | Marks |
|--------|----|---|---|-------|
| | ii | Use your array of authors from (i) to instantiate an object using the PhysicalBook class. | DECLARE OpenSecurity INITIALLY PhysicalBook("OpenSecurity", 12.99, myAuthors[], "Row C", 5) 1 mark for instantiation of PhysicalBook Class 1 mark for using all 5 values/instance variables | 2 |
| c | i | A total value for sales of an eBook is required so the eBook class is updated with a new method calculateSales(). (see paper) Using a programming language of your choice, write the code for line 152. | SET salesTotal TO THIS.QtySold * THIS.SalePrice 1 mark <i>Use of THIS is not mandatory</i> | 1 |
| | ii | The following line of code creates an array of 50 eBooks. (see paper) The store is interested in the total eBook sales value for all books with an English language author. Using a programming language of your choice, write the code used to calculate the total sales value for all eBooks where at least one author has a language of English. | 1 mark for traversing each available eBook object (FOR EACH loop or similar) 1 mark for traversing array of Author 1 mark for IF statement to identify "English" using of GetLanguage method 1 mark adding to total using calculateSales | 4 |
| | | | <pre> SET AllSales TO 0 FOR EACH soldBook IN eBooksSold DO SET english TO false SET size TO LENGTH(soldBook.Authors[]) SET authIndex TO 0 WHILE english = false AND authIndex < size DO IF soldBook.Authors[authIndex].GetLanguage = "English" THEN SET english TO true END IF END WHILE IF english = true THEN SET AllSales TO AllSales + soldBook.calculateSales END IF SET authIndex TO authIndex + 1 END FOR </pre> | |

| Number | | Question | Instructions | Marks |
|--------|-----|--|---|-------|
| | iii | Explain why it may have been more effective to use bubble sort than insertion sort for this function. | Bubble sort would have been able to complete the sort faster (1 mark) because it would have been able to move items into the correct position without the overhead of moving all the elements before moving the item into the correct position (1 mark) | 2 |
| 8 | | A TV Talent Show uses an audition management system which allows Assistants and Managers to register contestants and edit their personal details. Managers can also export statistics about contestants. Directors and Judges can view and edit personal details and audition records. Managers and Judges can generate reports and Judges can also setup callback auditions with contestants. | | |
| | a | With reference to the information above, draw a use case diagram to represent the requirements of the audition management system. | <p>1 mark 8 use cases with system boundary 1 mark for 3 actors 1 mark for extends or includes 1 mark for Association Links</p> | 4 |

| Number | | Question | Instructions | Marks | |
|--------|---|--|--|---|---|
| | b | <p>There are twelve seats in the semi-final of the TV Talent Show. As contestants are given scores out of 100, they fill the seats. Once all the seats are filled, the next contestant's score is compared with the lowest contestant in the seats. If the next contestant has a higher score, they replace the contestant with the lowest score, by taking their seat.</p> <p>The audition management system uses a record structure to hold the information about each contestant on a seat. The information stored about each contestant is: (see paper)</p> <p>The seats are laid out as follows in the TV studio. (see paper)</p> | | | |
| | | i | <p>Using a programming language of your choice, define a record structure to store the details for a contestant.</p> | <p>RECORD contestant { STRING name, BOOLEAN ParentPresent, INTEGER SemiFinalScore, INTEGER SeatNumber }</p> <p>1 mark</p> | 1 |
| | | ii | <p>Using a programming language of your choice, declare a 2-D array called ContestantSeating to store the details of contestants.</p> | <p>DECLARE ContestantSeating AS ARRAY OF ARRAY OF contestant INITIALLY [[]] <with 3 rows and 4 columns></p> <p>1 mark for correct 2D array. <i>Accept dimensions of [2][3]</i></p> | 1 |
| | | iii | <p>The twelve elements of the ContestantSeating 2-D array are initialised with the values "No Contestant", false, 0 and 0.</p> <p>Using pseudocode, design a sub-routine to initialise the variable ContestantSeating with the appropriate values.</p> | <p>1 mark for nest loops to process 2D array 1 mark for initialising correct values or records.</p> | 2 |

| Number | | Question | Instructions | Marks | |
|--------|---|--|--|--|---|
| | | | <pre> FOR rows = 1 TO 3 DO FOR column = 1 TO 4 DO SET ContestantSeating[row][column].Name TO "No Contestant" SET ContestantSeating[row][column].ParentPresent TO false" SET ContestantSeating[row][column].SemiFinalScore TO 0 SET ContestantSeating[row][column].SeatNumber TO 0 END FOR END FOR </pre> | | |
| | c | <p>Once all 12 seats are filled, each time a contestant is compared with the seated group of contestants an insertion sort is used to sort the scores and identify the seat number of the lowest scoring contestant.</p> <p>A function, called LowSeat, is used to return the seat number of the contestant with the lowest score. For input, the function receives two parallel 1D arrays; one of scores and one of seat numbers (both generated from the 2D array for the seated contestants) e.g. (see paper)</p> <p>The function makes use of the Insertion sort to sort both arrays in to ascending order of Score.</p> | | | |
| | | i | <p>Explain how the insertion sort algorithm would be used to place the value, 49, in correct position in the Score array.</p> | <ul style="list-style-type: none"> • 49 is out of place so the insertion sort works backwards down the array to identify the correct insertion position. • 49 is copied and the values from the insertion point upwards are moved up one place. • 49 is inserted at the insertion point. <p>1 mark each, max 3 marks</p> | 3 |
| | | ii | <p>Using pseudocode, write a solution for this function that will require the SeatNumber for the contestant with the lowest score.</p> | <p>1 mark for outer loop for length of array 1 mark inner loop to move items 1 mark for assignments to move items ups 1 mark for assignment and movement of temp values. 1 mark for returning the first element of array</p> | |

| Number | | Question | Instructions | Marks |
|--------|-----|--|--|-------|
| | | <p>FUNCTION LowSeat(ARRAY OF INTEGER Score, ARRAY OF INTEGER SeatNumber)</p> <p>FOR i = 0 to length(Score) DO</p> <p> SET ScoreTemp TO Score[i]</p> <p> SET SeatTemp TO SeatNumber[i]</p> <p> SET index TO i</p> <p> WHILE (index > 0) AND (ScoreTemp < Score[index-1]) DO</p> <p> SET Score[index] TO Score[index-1]</p> <p> SET SeatNumber[index] TO SeatNumber[index-1]</p> <p> SET index TO index - 1</p> <p> END WHILE</p> <p> SET Score[index] TO ScoreTemp</p> <p> SET SeatNumber[index] TO SeatTemp</p> <p>END FOR</p> <p>RETURN SeatNumber[0]</p> <p>END FUNCTION</p> <p><i>Other solutions are possible.</i></p> | | 5 |
| | iii | <p>Explain why it may have been more effective to use bubble sort than insertion sort for this function.</p> | <p>With bubble sort, the lowest value will be in position 0 (1 mark) but with insertion sort, it may take all passes for the lowest value to be in the correct position (1 mark)</p> | 2 |
| 9 | | <p>An online service is designed for a car rental company. The system is designed with the following entities and attributes. (see paper)</p> <p>Customers make a reservation for a type of vehicle which they will pickup and return to a location. Customers cannot make more than one reservation for the same date. When the customer arrives at the location, they are allocated a car of the required type. When the car is returned, the mileage the car has travelled and the amount of fuel in the tank is recorded as well as the date the car is returned.</p> | | |

| Number | Question | Instructions | Marks |
|--------|---|---|-------|
| a | <p>Draw an entity relationship diagram for this system. Show strong and weak entities and the participation between the entities. You do not need to include attributes in your diagram.</p> | <p>1 mark for each of the following, with correct cardinality and participation:</p> <ul style="list-style-type: none"> • Customer->Rental, Customer->Reservation • Vehicle->Rental, Location->Vehicle • VehicleType->Vehicle, VehicleType->Reservation <p>1 mark for both weak relationships</p> | 5 |
| | | | |
| b | <p>Customers need to know which types of vehicles are at each location. The customer will see the following information. (see paper)</p> <p>Write the SQL statement to display the number of vehicles, of each VehicleType, at this location.</p> | <pre>SELECT Location, Count(*) AS "At Location", Class, Seats, Doors FROM VehicleType, Vehicle, Location WHERE VehicleType.TypeID = Vehicle.TypeID AND Vehicle.LocationID = Location.LocationID AND LocName = "Dollar" GROUP BY VehicleType.TypeID</pre> <p>1 mark for correct fields 1 mark for correct equi-joins 1 mark for correct of COUNT and correct grouping.</p> | |

| Number | | Question | Instructions | Marks |
|--------|---|--|---|-------|
| 10 | | <p>A pizza delivery company allows orders to be placed online. (see paper)</p> <p>Orders are submitted via a form and processed securely by a server-side script called orders.php. The form begins with the HTML below. (see paper)</p> | | |
| | a | Complete the form element of the HTML code used to create this form. | <form method="post" (1 mark) action="orders.php" (1 mark)> | 2 |
| | b | <p>Some of the remaining HTML for this form is shown below. (see paper)</p> <p>The information returned from this form is used in the following INSERT statement to update the database.</p> | <pre>\$name = \$_post("name"); \$contactno = \$_post("contactno"); \$deliverytime = \$_post("deliverytime"); \$address = \$_post("address"); \$mealbundle = \$_post("mealbundle"); \$query = "INSERT INTO Order (Name, ContactNo, Address, DelTime, MealBundle) VALUES ('" . \$name . "', '" . \$contactno . "', '" . \$deliverytime . "', '" . \$address . "', '" . \$mealbundle . "')"; \$result = mysql_query(\$conn, \$query);</pre> <p>1 mark for use of \$_post to accept form values 1 mark for assignment of SQL statement to \$query 1 mark for use of PHP variables in SQL statement 1 mark for mysql_query using \$conn and \$query</p> | 4 |

| Number | | Question | Instructions | Marks | |
|--------|---|---|--|--|---|
| | c | <p>Early in the development progress, the development team completed usability testing using the following prototypes. (See paper)</p> <p>Explain how usability testing these prototypes would support the development of the pizza order system.</p> | <p>The development team will watch real users making use of the prototypes (1 mark)</p> <p>User feedback on:</p> <ul style="list-style-type: none"> • Placement of screen elements • Consistency of use of the interface • Controllable so users can backtrack • Helpful feedback to info user • All customisation for special user groups <p>1 mark for any two bullets of detailed user feedback, max 2 marks</p> | 3 | |
| 11 | | An online computer game uses a linked list to manage users who are waiting to gain access to the game server. (see paper) | | | |
| | a | The list is used to process users as follows. | | | |
| | | i | Describe the changes that will be made to the linked list when the node at the head of the list leaves to join the server. | <p>The head should be updated to 211 (1 mark) to remove 871 from the list (1 mark)</p> | 2 |
| | | ii | <p>A new user attempts to join the server but is added to the end of the list.</p> <p>User: RoroBear Address: 513</p> <p>Explain what must happen to add this user.</p> | <p>The list must be traversed until the NULL pointer is found at the end of the list (1 mark)</p> <p>The pointer (for node 712) is changed to 513, the value User: RoroBear is added to address 513, and the pointer at 513 is set to NULL. (1 mark)</p> | 2 |
| | b | i | <p>The makers of the online computer game want to launch a “Gold” service, where users, in any position, can pay a fee to jump up to 5 spaces closer to joining the server.</p> <p>Explain why this cannot be implemented with the current data structure.</p> | <p>This would require the list to be traversed to the end, and then traversal back up the list for a number of nodes. (1 mark)</p> <p>This is not possible because nodes (in a single linked list) are only connected in one direction. (1 mark)</p> | 2 |
| | | ii | Describe a solution, using a linked list, to the problem identified in (i). | <p>A double linked list has both next and previous pointers (1 mark) allowing the list to be traversed in both directions (1 mark)</p> | 2 |

Detailed Content

Section 1

| Question | Mark | Skills | Detail | Unit |
|----------|------|----------------|-----------------------------------|------|
| 1a | 1 | Implementation | Describe instantiation | SDD |
| 1b | 1 | Implementation | Encapsulation | SDD |
| 2 | 2 | Implementation | CSS Media Queries | WDD |
| 3a | 2 | Implementation | PHP: mysqli_connect | WDD |
| 3b | 1 | Implementation | PHP: mysqli_close | WDD |
| 4 | 4 | Implementation | SQL: CREATE, primary/foreign keys | DDD |
| 5 | 2 | Testing | Integration testing | SDD |
| 6a | 1 | Implementation | PHP: Sessions | WDD |
| 6b | 1 | Implementation | PHP: start_session | WDD |

15 marks

Section 2

| 7a | 2 | Implementation | Implement sub-class | SDD |
|-------|---|----------------|-----------------------------------|-----|
| 7ai | 3 | Implementation | Array of Objects | SDD |
| 7bii | 2 | Implementation | Instantiation of an object | SDD |
| 7ci | 1 | Implementation | Property | SDD |
| 7cii | 4 | Implementation | Method, Class | SDD |
| 8a | 4 | Analysis | Use Case Diagram | SDD |
| 8bi | 1 | Implementation | Record structure | SDD |
| 8bii | 1 | Implementation | 2D array of records | SDD |
| 8biii | 2 | Design | 2D array, record handling | SDD |
| 8ci | 3 | Implementation | Insertion Sort | SDD |
| 8cii | 5 | Design | Insertion Sort | SDD |
| 8ciii | 2 | Implementation | Bubble sort, Insertion Sort | SDD |
| 9a | 5 | Design | ERD, strong, weak, participation. | DDD |
| 9b | 3 | Implementation | SQL: SELECT, COUNT, GROUPING | DDD |
| 10a | 2 | Implementation | HTML: form element | WDD |
| 10b | 4 | Implementation | PHP: \$_post, msqi_query | WDD |
| 10c | 3 | Testing | Usability Testing with Prototypes | SDD |
| 10di | 2 | Implementation | Single linked list | SDD |
| 10dii | 2 | Implementation | Single linked list | SDD |
| 10ei | 2 | Implementation | Single linked list | SDD |
| 10eii | 2 | Implementation | Double linked list | SDD |

55 marks