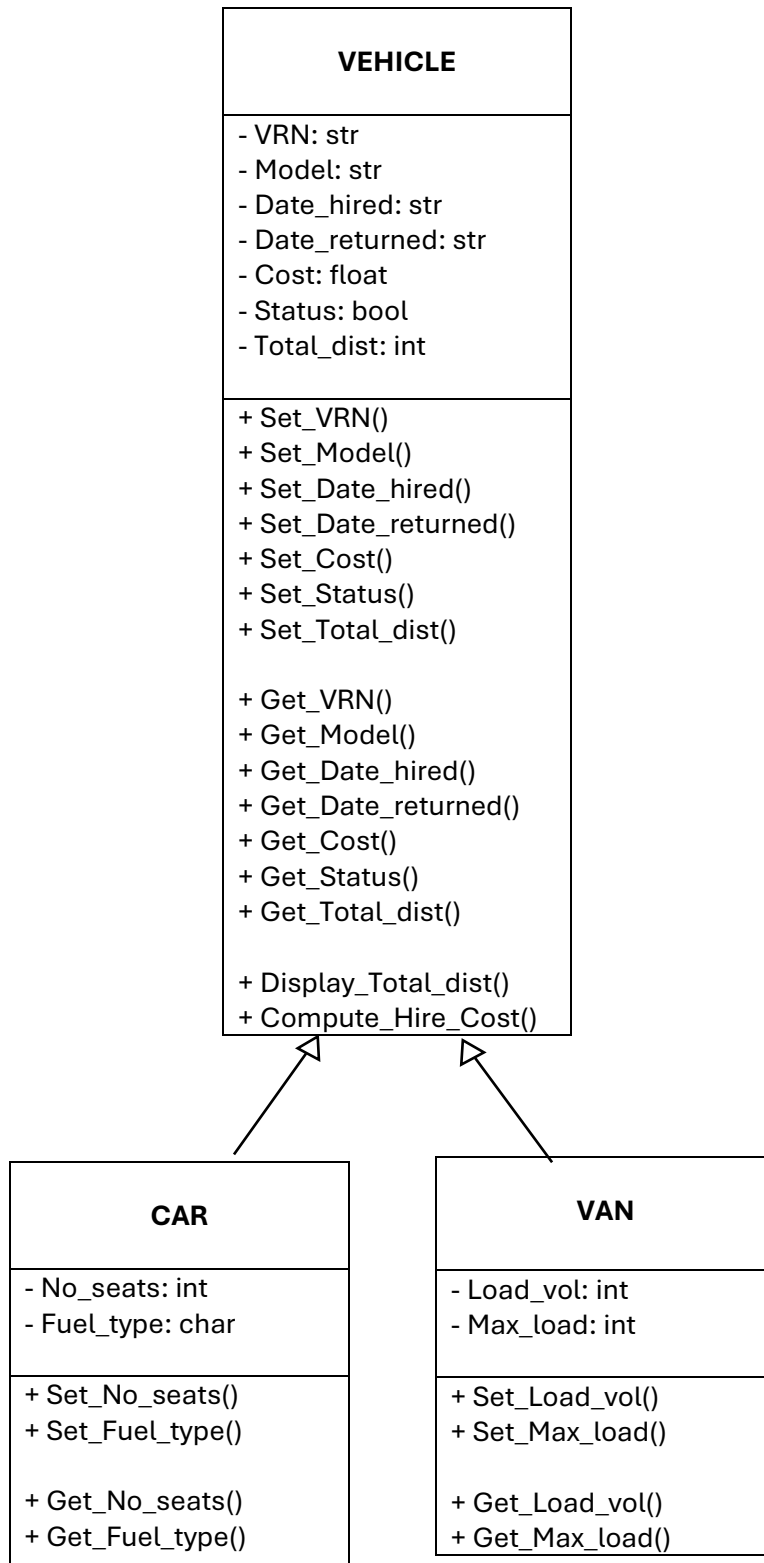


H2 Computing 9569/01 2020 Suggested Solutions

1(a) 12m



1(b) 2m

The purpose of a superclass is to allow reusable code by allowing subclasses to inherit its attributes and methods and to extend it further.

The superclass VEHICLE contains attributes that subclasses CAR and VAN inherits with each having its own additional attributes such as No_seats for CAR and Load_vol for VAN.

1(c) 2m

Encapsulation is the bundling of data (attributes) and functions (methods) that operates on these data into packages called classes.

This allows information hiding to be implemented where programmers can only use the given public interface (methods) to access the private attributes. This reduces accidental errors as programmers cannot change the attribute directly which may lead to an inconsistent state. Instead, implementation details are hidden from the programmer using the class. This also promotes usability as encapsulation allows access to a level without revealing the complex details below that level.

Encapsulation also allows for easier maintenance of code as code changes can be made independently.

1(d) 1m

In the CAR class diagram, a public method Compute_Hire_Cost() must be added to indicate that this method will behave differently from the method of the same name in VEHICLE as it uses a different formula to compute the hire cost.

1(e) 1m

Purpose of Polymorphism is to promote code extensibility as new types of objects that can respond to existing messages (e.g. Compute_Hire_Cost) can be added without modifying the base object.

2(a)(i) 1m

The ideal pivot for the quicksort algorithm to execute most efficiently is to choose a pivot that will result in two sub-arrays of equal size after the first pass. So if the data set is an unsorted array of integers the **median value** of the integers should be chosen as the pivot.

2(a)(ii) 1m

The median value of the data set can be in any random position of the array as it is unsorted. So, a linear search has to be done to find the location of the ideal pivot.

2(b) 2m

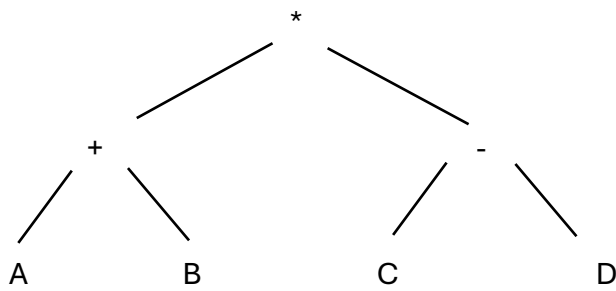
For data sets that are already sorted, nearly sorted, or reverse sorted, choosing the first or last position of the pivot will lead to the worst-case $O(n^2)$ time. If random selection is used instead, the chance of selecting a worst-case pivot is largely reduced. On average, this will result on better performance over selecting first or last position as the pivot.

2(c) 4m

For already sorted or nearly sorted data sets, insertion sort has a time complexity of $O(n)$ vs a divide and conquer algorithm like quicksort. This is because in a largely sorted array, the number of comparison of insertion sort is reduced as the comparison stops once a number is found to be smaller. It only require a few $O(n)$ passes to insert the out of order elements. For quicksort, the number of comparisons to be made largely depends on the pivot selected giving a best case of $O(n \log n)$ and worst case of $O(n^2)$.

Also, since insertion sort is a stable sorting algorithm, in sorting a nearly sorted array, the overheads are also largely reduced compared to the quicksort algorithm which is an unstable sorting algorithm.

3(a) 2m



3(b) 1m

Index of a leaf node is 0 (or 2 or 4 or 6)

3(c)(i) 1m

Recursion is a function that is defined in terms of itself. It calls itself with one or more similar but smaller problems until one or more terminating case(s) is reached.

3(c)(ii) 1m

Line 08 is the terminating (base) case.

Lines 03 and 06 are the recursive calls.

3(c)(iii) 1m

Significance of line 02 and 05 is to check if a leaf node has been reached. If it is not a leaf node (index is not -1) then it will do a recursive call on the left or right subtree to continue traversing the tree.

3(d) 5m

Index	Output
1	
0	A
2	B
	+

(Note to students: recursive calls are not output. In procedure P, only line 8 is the output)

3(e) 3m

When procedure P is called, memory is set aside to store local variables as well as information about the procedure such as its address in memory which will allow the program to return to the proper place after a procedure call. This memory area is known as a frame.

Every time a recursive call happens, a new frame for that call will be created and pushed to a stack called the call stack. As a stack is a LIFO (last in first out) data structure, the last function to be called (the most recent one) will be pushed to the top. When it finishes, its frame is destroyed and removed from the stack, returning control to the frame just below it on the stack (the new top frame). So a stack is used to keep track of the order of recursive calls and its correct return address.

3(f) 1m

Procedure P performs a post order tree traversal (Left Right Root).

3(g) 2m

A queue is a First In First Out (FIFO) data structure. Items are inserted at the end/back of the queue (Enqueue operation) while the first item (head of the queue) in the queue will be removed first (Dequeue operation).

3(h) 2m

	Linear Queue	Circular Queue
1.	Arranges the data in a linear pattern.	Arranges the data in a circular order where the rear end is connected with the front end.
2.	The insertion and deletion operations are fixed i.e, done at the rear and front end respectively.	Insertion and deletion are not fixed and it can be done in any position.
3.	In the case of a linear queue, the element added in the first position is going to be deleted in the first position. The order of operations performed on any element is fixed i.e., FIFO.	In the case of circular queue, the order of operations performed on an element may change.
4.	Front elements cannot be reused once a dequeue operation takes place.	Space in front can be reused when the tail wrap around to the front.

4(a)(i) 2m

IP (Internet Protocol) address and MAC (media access control) address are two ways a device can be identified on a LAN.

4(a)(ii) 2m

Communications protocol is a standard set of rules used to ensure the proper transfer of data between devices. Protocols exist that specify the format of the data, and the signals to start, control and end the transfer. Without protocols, devices will be unable to initiate and to facilitate the exchange of data between them.

4(b)(i) 3m

Packet switching is a connectionless network switching technique. The data is divided and grouped into a number of units called packets that are individually routed from the source to the destination. There is no need to establish a dedicated circuit for communication.

Each packet in a packet switching technique has two parts: a header and a payload. The header contains the addressing information of the packet and is used by the intermediate routers to direct it towards its destination. The payload carries the actual data.

A packet is transmitted as soon as it is available in a node, based upon its header information. The packets of a message are not routed via the same path. So, the packet in the message arrives in the destination out of order. It is the responsibility of the destination to reorder the packets to retrieve the original message.

4(b)(ii) 3m

As the movement of packets in packet switching is not synchronous, it may not be suitable in communication applications like voice calls as time delays may occur in waiting for out of order packets and reassembling of the message. Using circuit switching will solve this issue. However, in packet switching, this problem can also be minimized by additional data compression (speeding up transmission time) or using Quality of Service (QOS) in routers where priority is given for these packets for quicker delivery to the destination.

4(b)(iii) 2m

In packet switching, the packets can take one or many paths via the different nodes to reach the destination. Thus, if one node is not operational due to a broken cable, the

router will simply select an alternative route for the packet to travel via another node to the destination.

4(c) 4m

Step 1: The client proposes a domain name resolution request and sends the request to the local domain name server.

Step 2: When the local domain name server receives the request, it first queries the local cache. If there is this record, the local domain name server directly returns the result of the query.

Step 3: If the local cache does not have the record, the local domain name server directly sends the request to the root domain name server, and then the root domain name server returns the primary domain name of the domain (the subdomain of the root) of the local domain name server.

Step 4: The local server sends a request to the domain name server (Top Level Domain server) returned in the previous step, and then the server that accepts the request queries its own cache. If there is no such record, it returns the address of the relevant lower-level domain name server (Authoritative Server).

Step 5: Repeat step 4 until you find the correct record.

Step 6: The local domain name server saves the returned results to the cache for the next use and returns the results (the webpage ip address) to the client.

4(d)(i) 1m

Use of encryption and decryption. Public key cryptography such as RSA – use of public key to **encrypt data** that receiver can **decrypt** only with a private key.

4(d)(ii) 1m

Use of a digital signature – sender uses a private key to encrypt the data that can only be decrypted using the known public key showing proof of origin.

5(a)(i) 1m

Purpose of verification in data entry is to make sure that data entered is the same as the source.

5(a)(ii) 1m

One method of verification is double entry verification to ensure data typed into a computer system is entered accurately. The data is entered twice, by different operators, and compared by the system. Any differences can be identified and manually corrected.

5(b)(i) 1m

Validation involves using the properties of the data to identify any inputs that are obviously wrong. Validation only proves that the data entered is a reasonable value for the computer to accept. It cannot prove that the data entered is the actual value the user intended. However, it does allow the computer to filter out obvious mistakes.

5(b)(ii) 3m

Range checks

Presence checks

Length checks

Type checks

Format checks

(any 3 of the above. Note: parity check or checksum is not correct in the context of data entry. It is only correct for the context of data transmission.)

5(b)(iii) 2m

Transcription and Transposition errors.

5(c) 3m

Step 1:

$$0*6 + 2*5 + 7*4 + 5*3 + 7*2 = 10 + 28 + 15 + 14 = 67$$

(multiply each digit by its weight in that position. Find the sum)

Step 2:

$$67 \% 11 = 1$$

(Take the sum and find the result of modulus 11)

Step 3:

$$11 - 1 = 10$$

(Subtract the remainder from 11 to get the check digit)

Step 4:

The check digit cannot be 10 (**2 digits**) so it is replaced with X (**or any chosen single character**).

Answer: X

5(d) 2m

One reason is that the check digit computed is 10 which needs to be replaced with a single symbol (character). Hence, it needs to be stored as a string.

Another reason is that if it is not stored as a string, the leading zero would be discarded. So, to maintain the correct number of digits including any leading zero, it should be stored as a string.

6(a) 2m

The table is not in 1NF as there are attributes that do not contain atomic values. For example under one particular Student ID, there are several different values for Course ID, Subject, Teacher ID, Teacher Name and Room number.

6(b)(i) 1m

Suitable Primary Key: Student ID

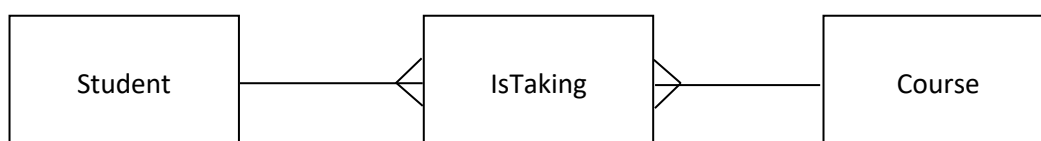
6(b)(ii) 1m

Suitable Primary Key: Course ID

6(b)(iii) 1m

Suitable Primary Key: Student ID + Course ID (**composite key**)

6(c) 3m



6(d) 2m

Table Course is not in 3NF because it has transitive functional dependency between its fields. Course ID determines Teacher ID which determines Teacher Name. Therefore, Course ID determines Teacher Name via Teacher ID.

6(e) 7m

Student(Student ID, First Name, Last Name)

Course(Course ID, Teacher ID, Subject)

IsTaking(Student ID, Course ID)

Teacher(Teacher ID, Teacher Name, Room Number)

6(f) 2m

Reducing data redundancy reduces insert, update, and delete anomalies leading to data inconsistencies compromising data integrity. It also saves memory space as duplicated data are removed potentially enhancing the efficiency of the database since it is smaller in size.

6(g) 5m

```
SELECT Subject, Teacher Name, Room Number
FROM Teacher
INNER JOIN Course
WHERE Teacher.Teacher ID = Course.Teacher ID
AND Course.Course ID = (SELECT Course ID
                        FROM IsTaking
                        Where Student ID = 1395)
ORDER BY Subject ASC
```