2023 A-level H2 Computing Paper 1

1 (a)(i) For a table to be in third normal form, the table should be in second normal form and it should not have transitive dependencies - all fields must only be determined by the primary/composite key, not by other non-key attributes. The table Car is not in third normal form because DayRate is dependent on Category, which is a non-key attribute in the table.

(ii) There will be a case of data redundancy as cars of the same Category will require the same DayRate to be entered into the table more than once which can in turn lead to data inconsistency if an insert, update or delete error is made.

(b)

Car(RegistrationNumber, Make, Model, CategoryName*)

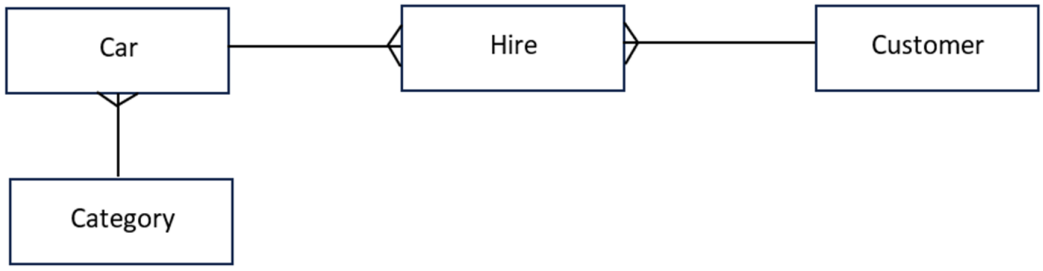
Category(CategoryName, DayRate)

Customer(DriverLicenceNumber, Name, Address, TelephoneNumber)

Hire(RegistrationNumber*, DriverLicenceNumber*, DateHired, DateExpectedBack, DateReturned)

* denotes foreign key

(c)



(d)(i) 1. User can be requested to enter licence number two times (double entry)

2. User can be prompted to proofread entry after the licence number is entered.

(ii) Presence check: can be used to ensure that data expected back is not left blank.

Range check: can be used to ensure that cars are hired for a minimum period of one day (For example, DateExpectedBack cannot be a date earlier than DateHired.)

Format check: can be used to ensure that the data expected back is entered with a correct required format, for eg DDMMYY.

(e)

Advantages

- Offline Functionality: Native apps can often provide a certain level of functionality even when there is no internet connection. They can cache data locally and allow users to access certain features or content offline
- Performance: Native apps are optimized for specific platforms and can leverage the full capabilities of a device, resulting in faster and more responsive performance.
- Access to Device Features: Native apps have direct access to device features like camera, GPS, and sensors, allowing developers to create more feature-rich and interactive applications.

Disadvantages

- Limited Access to Device Features: Web apps have restricted access to certain device features compared to native apps, limiting their ability to utilize certain hardware capabilities.
- Dependency on Internet Connection: Web apps often require an internet connection, and their performance can be affected by network conditions, potentially leading to a suboptimal user experience in offline situations.

2 (a) There is no need to allocate or deallocate memory during execution with faster access to data and prevent memory leaks.

It avoids memory fragmentation as memory blocks are contiguous and aligned.

(b) With dynamic allocation, the size of the data structure can be adjusted during runtime based on actual requirements. This flexibility is particularly useful when dealing with data structures whose size is not known at compile-time or varies dynamically.

It allows for efficient memory utilization, preventing the allocation of unnecessary memory and accommodating changing data needs.

(c) Static memory allocation is more appropriate for arrays. An array contains a fixed number of items and requires items to be stored in contiguous memory location, both of which are afforded by static memory allocation.

Comments:

- Static memory allocation is more appropriate when the size is fixed. For example, storing data from the periodic table.
- If the data size is small and fixed, then static memory allocation is more appropriate since access is easier and faster.

(Need not include data structure for c since that is in part d)

(d) Queues also benefit from dynamic memory allocation as we can make use of pointers to enqueue and dequeue instead of the more expensive method of removing an item and shifting the rest of the items forward if static memory allocation was used.

Comments:

- Binary Search Tree because it is a variable sized, hierarchical structure.
- When using BST, size of nodes is unknown whereas size of the queue is known?
- Will it be more efficient to balance a BST when using dynamic memory allocation?

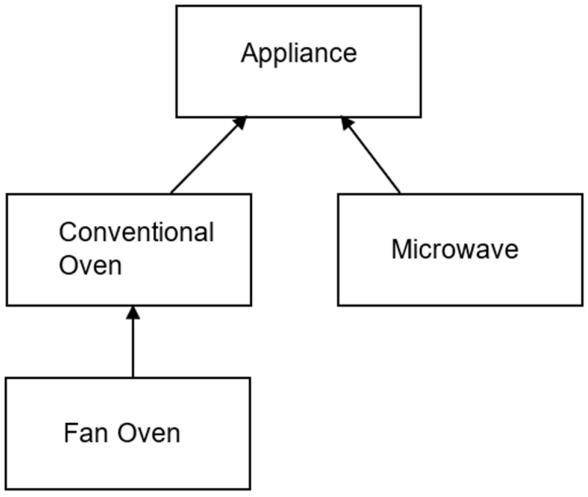
(e)

```
FUNCTION find(LinkedList: ARRAY, Value: REAL, head: INTEGER) RETURNS INTEGER
	Current <- head
		WHILE Current  < > - 1 DO
		IF LinkedList[Current].data = Value
			RETURN Current
		ELSE
			Current <- LinkedList[Current].next
		ENDIF
	ENDWHILE
	RETURN -1
ENDFUNCTION
```

(f)

```
FUNCTION push(Stack: ARRAY, Value: REAL, head: INTEGER) RETURNS BOOLEAN
	IF top < 100
		top <- top + 1
		Stack[top] <- Value
		RETURN True
	ELSE
		RETURN False
	ENDIF
ENDFUNCTION
```

3 (a)



(b) Attributes:

```
status (on/off)
timer
alarm
```

Methods:

```
turn_on()
turn_off()
set_timer()
start_timer()
end_timer()
start_alarm()
stop_alarm()
```

(c) It ensures attributes are only accessed/altered by calls to public methods. This helps to maintain data integrity, ensuring modifications are done deliberately.

Subroutines (used within the class itself/ is sensitive) invoked by public methods can be private.

Comments:

- Private methods allow a class to control its interactions with the outside world. This control ensures that critical operations that should not be exposed to external use remain hidden. This is especially important for methods that perform sensitive operations or handle critical data, where incorrect usage could lead to bugs or security vulnerabilities.

(d) Polymorphism allows methods in the subclass to have the same name as in the superclass to allow for different behaviour.

4(a) Elements in an array can be retrieved by index with a constant access time O(1), but a linked list requires iteration/ traversal. Time complexity is O(n) for a linked list with n items for the worst case scenario as a linked list requires traversing through the entire list to search for the item.

Comments:

- Assume the array contains data in ascending order.
- Assume a classical linked list with only head pointer and link/next references are one way only.

Advantages of storing ordered data in array vs linked list

- More efficient search using binary search O(lg n) (prerequisite: sorted data) compared to linked list using linear search O(n).

- Since the data is ordered, we may apply a binary search algorithm on an array instead of a linear search on a linked list. Hence, the time complexity for search is $O(\log_2 n)$ instead of $O(n)$.
- But should we compare using two different search algorithms?
- More versatile traversal: can process data in both ascending (count up loop) and descending order (count down loop) using array indices directly; not possible to traverse both ways using classical linked list
- Consider ordered data context for both advantages instead of generic pros of array (constant time direct/random access).

4(b)

Place the first data item in the root node.

For each subsequent data item in turn, start comparing at the root node each time. If the data item is less than data in the root node, branch to the left, and if it is greater than the data in the root node, branch to the right.

Apply the rule at each node encountered until an empty branch is found. Add data as a child node at the empty branch.

This continues until all data items are added into the binary search tree.

Comment: Alternatively, use recursion.

Base case: Empty tree. Insert a new node as root.

Recursion: Compare new node's value to root. If the new node's value is smaller, go to the left child if it exists and insert the new node into the subtree that starts from the left child. (Similar sentence if new node is bigger than root)

(c) The shape of the binary search tree may differ when the order of how the data items are added and changed.

(d)(i) Enqueue and dequeue

(ii) Front pointer denoting start of queue and rear pointer denoting end of queue.

(iii) It can be applied in a printer server holding print jobs/ song playlist/ online tickets queue/ login servers for online gaming/ reservation systems at restaurants/ breadth-first-search for trees.

Depth-first-search using stack.

(iv) In a linear queue, when elements are dequeued, the front positions become empty and cannot be reused until the entire queue is empty and reset. This leads to potential wastage of space.

Circular queues address this issue by allowing the front and rear to wrap around, enabling the reuse of the space vacated by dequeued elements without the need to reset the entire queue. As a result, circular queues can be more memory-efficient.

5(a) (i) The network application uses a hash algorithm to create a hashed version of the message along with a private key to encrypt the hash to the digital signature.

(ii) Kim's network application uses the sender's public key to decrypt the digital signature back to the sender's version of hash. Then, it uses the same hash algorithm to create a new hash from the received message. If the two hashes match, the data was not altered and the digital signature is authentic.

(b) (i) The address consists of 32 bits, written in dotted decimal notation with 4 decimals separated by dots.

(ii) It can be allocated dynamically when the host joins the network via Dynamic Host Configuration Protocol (DHCP) or persistently by configuration of host hardware or software.

State dynamic vs static IP address allocation

(iii) In a packet-switching network, data is divided into small segments known as packets. These are divided so that each packet forms part of a complete message that can be routed through a network of switches to its destination independently of all other packets forming the same message.

Make it clearer that there are different possible paths (that are not set in advance), and the packets may arrive out of order.

(iv) A router connects different links. The role of a router is to transmit an incoming packet from one input link to an outgoing link. It examines IP packet headers, looking at the destination address and consulting a routing table of known networks. The table indicates which connections to send the packet on as the next link. In the event that the entire packet has not been sent, it will buffer or store the packet's bits and only transmit when all the packet's bits have been received.

A router allows for internetwork communications. It forwards the packets to the destination IP address using the most efficient/ best path.

(c) (i) 77

(ii) 01001101

(iii) UTF-8 can represent more symbols, characters and languages when compared to ASCII as it extends to 8-bit codepoints as compared to the 7-bit codepoints for ASCII.

Unicode - 1 to 4 bytes to represent a character

ASCII - 1 byte to represent a character

6 (a) Choose any element to be the pivot. The algorithm then partitions the array around the pivot, putting every element less than the pivot in a less_than array, and every element greater than and equal to the pivot in a greater_equal array. The equal array will just hold one element – the pivot itself. The process repeats for the less_than array and greater_equal array recursively. Base case of 1 or 0 element. Eventually, the elements are combined in order by first inserting the elements of less_than, then the pivot, and finally those of greater_equal.

Include the given data in the explanation. For example, Choose the first element (10) as the pivot. Read and compare the unsorted elements with the pivot and partition into two subarrays…

In-place Quicksort algorithm for theory and not in-place algorithm for programming if not specified.

(b) $O(n^2)$

(c) (i) It is when a function calls itself repeatedly to solve a smaller sub-problem until the base case is met.

(ii)
```
FUNCTION recursive(number1)
    total = 10
    IF number1 = 0
            RETURN total
    ELSE
            RETURN recursive(number1 - 1) + 3
    ENDIF
ENDFUNCTION
```

7 Ensure that customers have given consent to the collection of data.
Ensure that customers are aware of the purpose of collecting, using or disclosing the data.
Ensure that upon request, customers will be notified on how the personal data has been used or disclosed within a year unless the disclosure is expected to cause harmful effects.

Ensure that the data is accurate and complete.

Ensure that the data is reasonably secured to prevent unauthorised access, collection, use, disclosure or similar risks.

Cease retention of data if it is no longer necessary.

8 No, the manager is not. By delivering a system with a known bug that may have catastrophic consequences, it can potentially harm people's lives. While it is admirable to deliver the system on time such that it does not interfere with other work, it is important to consider the social consequences.

Include the nuclear power station example in the discussion.

Timeliness, need for prototype, law abiding

1. GENERAL ETHICAL PRINCIPLES.
A computing professional should...

1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.
1.2 Avoid harm.
1.3 Be honest and trustworthy.
1.4 Be fair and take action not to discriminate.
1.5 Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.
1.6 Respect privacy.
1.7 Honor confidentiality.

2. PROFESSIONAL RESPONSIBILITIES.
A computing professional should...

2.1 Strive to achieve high quality in both the processes and products of professional work.
2.2 Maintain high standards of professional competence, conduct, and ethical practice.
2.3 Know and respect existing rules pertaining to professional work.
2.4 Accept and provide appropriate professional review.
2.5 Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.
2.6 Perform work only in areas of competence.
2.7 Foster public awareness and understanding of computing, related technologies, and their consequences.
2.8 Access computing and communication resources only when authorized or when compelled by the public good.
2.9 Design and implement systems that are robustly and usably secure.

3. PROFESSIONAL LEADERSHIP PRINCIPLES.
A computing professional, especially one acting as a leader, should...

3.1 Ensure that the public good is the central concern during all professional computing work.
3.2 Articulate, encourage acceptance of, and evaluate fulfillment of social responsibilities by members of the organization or group.
3.3 Manage personnel and resources to enhance the quality of working life.
3.4 Articulate, apply, and support policies and processes that reflect the principles of the Code.
3.5 Create opportunities for members of the organization or group to grow as professionals.
3.6 Use care when modifying or retiring systems.
3.7 Recognize and take special care of systems that become integrated into the infrastructure of society.

4. COMPLIANCE WITH THE CODE.
A computing professional should...

4.1 Uphold, promote, and respect the principles of the Code.
4.2 Treat violations of the Code as inconsistent with membership in the ACM.