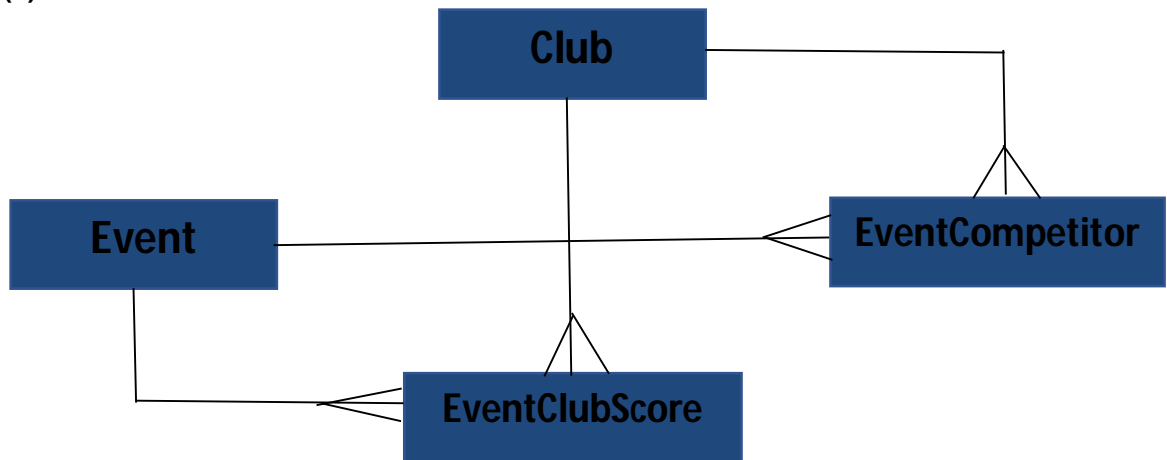


**LWL**

**1(a)**



**1(b)**

Club (ClubName, CS\_Name, CS\_Email)

Event (EventNum, Date, StartTime, Location)

EventCompetitor (ClubName\*, CylistName, CompetitorNum, EventNum\*, FinPosn, FinTime)

EventClubScore (ClubName\*, EventNum\*, ClubScore)

**1(c)**

SELECT Club.ClubName, EventClubScore.ClubScore, Club.CS\_Name, Club.CS\_Email

FROM Club, EventClubScore

WHERE EventClubScore.EventNum = 23 AND

Club.ClubName = EventClubScore.ClubName

ORDER BY EventClubScore.ClubScore DESC

**1(d)**

Normalisation aims to reduce data redundancy and minimise data inconsistency.

Two other requirements are 1) no partial key dependency and 2) no non-key dependency.

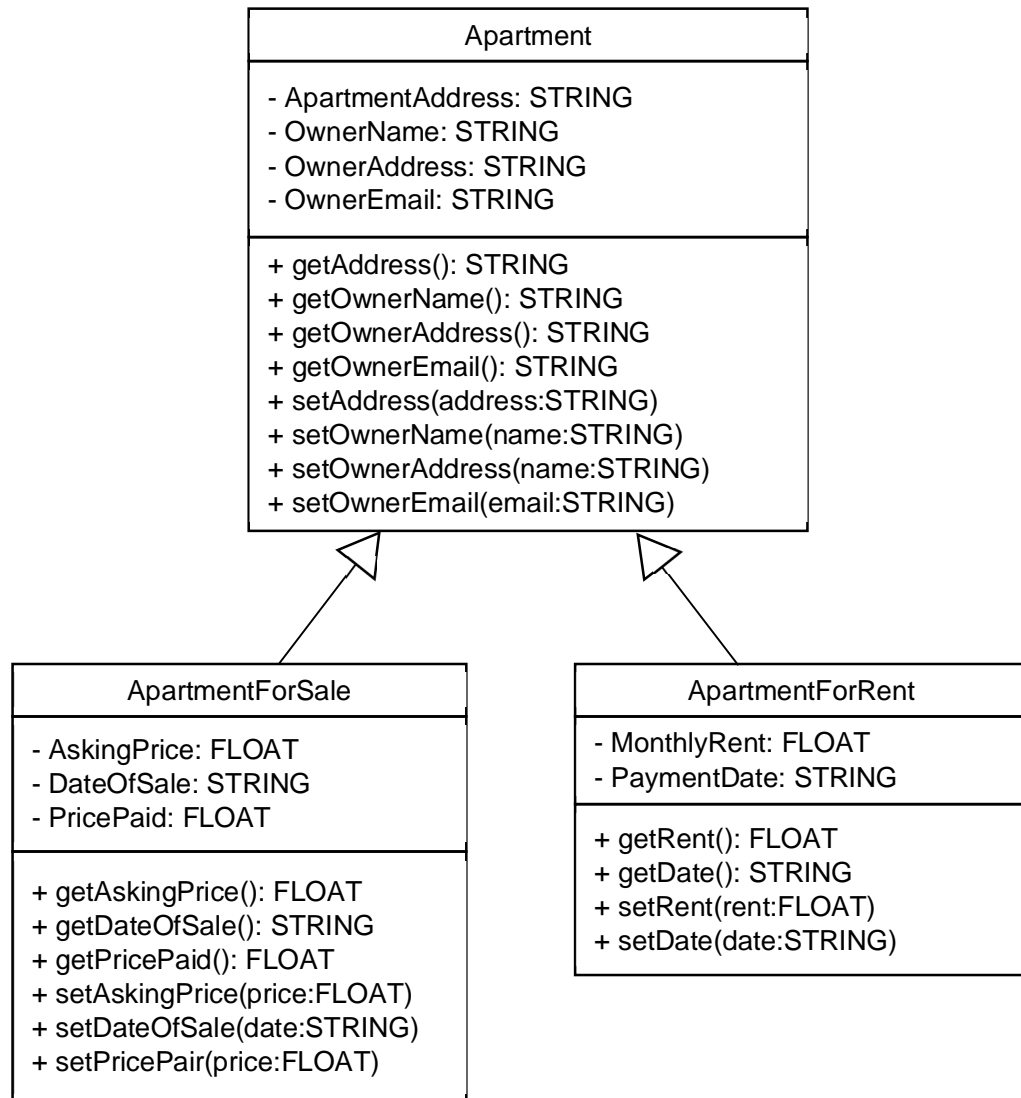
**1(e)**

Range check between 0 to 25 inclusive.

**CJJ**

- 2 (a) A class is a template for an object, and it defines the attributes and methods of objects in that class. An object is an instance of a class.

(b)



- (c) This prevents classes from seeing or changing the properties of other classes, even when they can access the methods belonging to these classes. This will protect the integrity of the properties in the classes.
- (d) It allows the developer to reuse the shared attributes and methods in one class for another class, and it allows him to save time from having to type the same code for multiple classes.

**3 (a)** This can be done through encrypting the message that only the intended recipient can decrypt to read it.

**(b)** The sender can use a hash algorithm on the message to be sent to create a hash value. This hash value will be encrypted with the sender's private key, and then attached to the message and sent as the digital signature. The sender then provides the receiver with the signed message and the corresponding public key for decrypting the signature to reveal the hash value. The receiver then uses the same hash algorithm to create a new hash value on the message text. If the two hash values match it will prove that the data has not been altered in the process of transmission.

If there is any malicious alteration of the message during its transit, the signature will be changed, causing the receiver's hash value to not match the sender's hash value, thus preventing the message from being decrypted.

**(c)** Firstly, authentication by the server is important in ensuring that the client accessing data belonging to an account in the server is authorized to do so. This can be achieved through password protection and multi-factor authentication.

Secondly, authentication by the client is important in ensuring that the server is the system that it claims to be. This can be done by the server issuing a certificate from a trusted third party, such as Transport Layer Security certificates, to the client.

## **LWL**

**4(a)**

Fast access of data, and less memory needed to store each item

**4(b)**

Fast insertion and deletion of data, and no limit to the capacity.

**4(c)**

Line 06 – It is a recursive function.

Lines 02 and 03 – Base case

Function Z returns the number of nodes in the linked list.

**4(d)**

**4(e)**

If the choice of pivot and order of data results in an unbalanced tree for quicksort, it will lead to the worst-case time complexity. That being the case, the worst-case time complexity of merge sort is  $O(n \log_2 n)$  while the worst-case time complexity of quick sort is  $O(n^2)$ .

**CN**

**5(a)**

The issue with this algorithm that runs on dynamic memory allocation happens whenever a user enters the lift and at level  $n$ , and chooses level  $n$  lift button of level similar to the level he/she enters, memory will be requested and then dynamically allocated to store the value  $n$ . Since level lift is already on the level  $n$ , the algorithm will complete (Finished) without releasing the memory requested to store value  $n$  back to the memory pool (heap), will wait for the next button press by a user.

In dynamic memory allocation, all memory allocated on the heap, must be freed at the end of the program, or whenever it is no longer needed in order to prevent memory leak. Even the likelihood of a person pressing on the chosen floor button that is the similar to the floor level he/she enters may be low, a cumulation of a substantial number of such instances happen and over a period of time will result in more and more memory allocated without being released back to the memory pool.

**5(b)**

Release the memory used to store the floor number if the elevator is already at the chosen floor so that memory will be deallocated.

**5(c)**

The mistake was not identified during testing because the programmer tested his codes by consideration of a set of valid input values. This would not result in any run-time error. Fault switches may not be able to detect logic errors, during testing the original algorithm will not immediately trigger off any fault switch. This led him to think that his algorithm is correct.

**CJJ**

**6 (a) (i)** It allows users to keep additional copies of their data which they can recover should the primary version of the data is lost or corrupted.

**(ii)** Archiving allows data to be stored for long-term retention, while helping to manage hardware space limitations.

**(b)** This is to prevent the backup copies of the data from being lost due to hardware damage at the primary site, due to threats such natural disasters.

**(c)** The business can lose all its critical information (such as sales records and inventory records) due to hardware failure or data corruption. The business will have to spend extra time to come up with all the information again, which can lead to potential loss of profit and income. This can also lead to loss of trust in their clients.

**CN****8**

The average case scenario is mainly used in the consideration of performance comparison of different search methods.

By considering a hash table that is sufficiently large enough with a properly defined hash function that minimises the occurrences of collisions, hash table search will take  **$O(1)$**  time **in most cases** to look up for the address of a record using its key, and then **directly access** the record in the memory. On the other hand, linear search and binary search will take in most cases  $O(N)$  time and  $O(\log_2 N)$  time complexity respectively.

In addition, when comparing hash table search and binary search, it can be argued that every comparison adds costs as the binary search makes will require data to be read data, and if it is not found, it will need to determine the next index to read, where as a hash table search allows direct access to the memory at the address that was hashed.

Given that  $O(1) < O(\log_2 N) < O(N)$ , we can conclude that the hash table search has a considerable advantage over binary search and linear search.

- b. A collision in a hash table happens when the keys of two or more distinct records are hashed to the same address location in the memory.
- c. Chaining can be used to handle the consequence of a collision. In chaining, records are stored in a node of one of the many singly-linked lists. Each array item of the hash table will store the address of one singly-linked list. Whenever a key of a record gets hashed to an address where another record has already been hashed to, the new record will be added to the end of the list that its hashed address points to.
- d. Features of an effective hash algorithm include:
  - Quick computation of the hash address.
  - Minimal occurrences of collisions / Maximise range of distinct output values.
  - Uniform distribution output ie. every distinct hash value that can be produced by the hash function has an equally likelihood of being chosen.