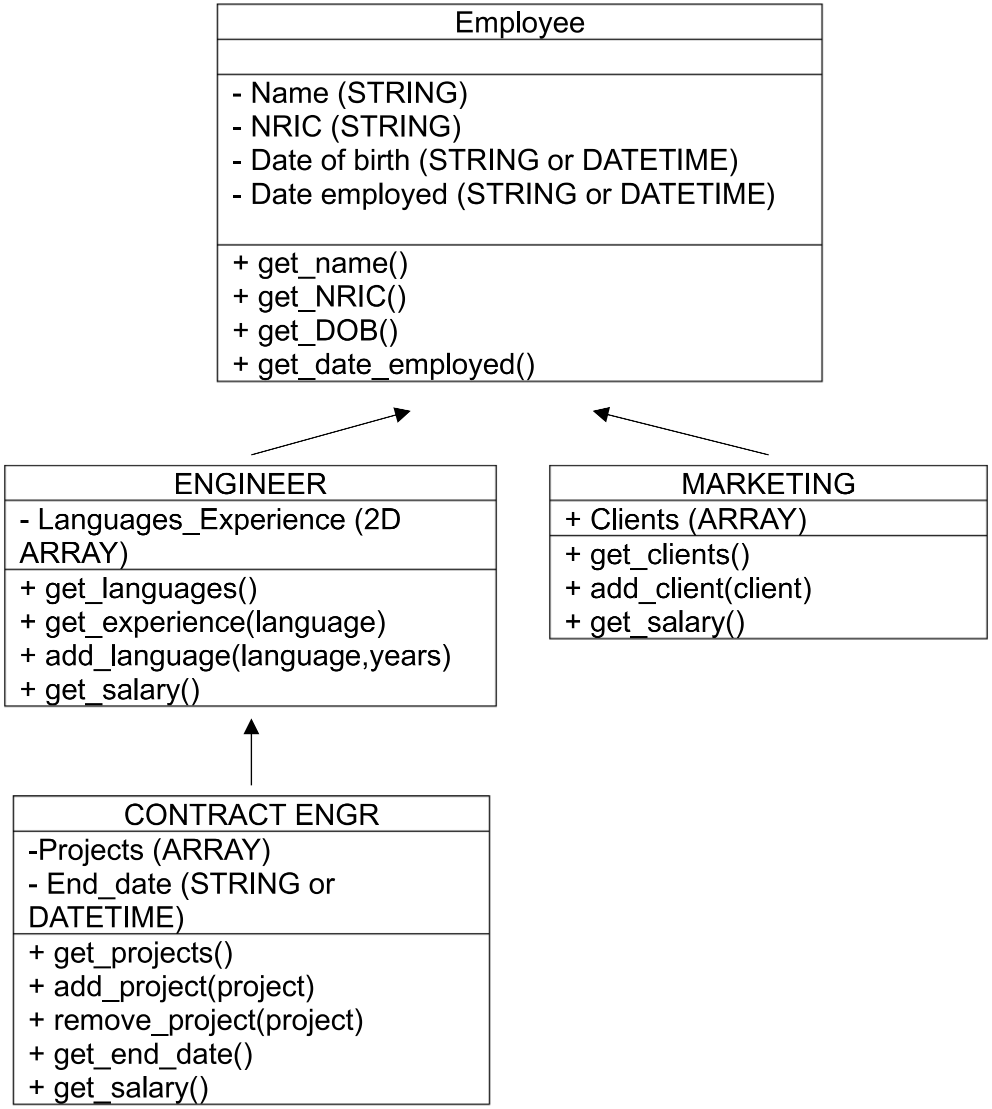
<table>
<tr><td>1(i)</td><td>$48_{10} = 110000_2$
$57_{10} = 111001_2$

The last 4 digits interpreted as a binary number convert to the base 10 digit they represent ($0000_2 = 0_{10}$, $1001_2 = 9_{10}$).</td><td></td></tr>
<tr><td>(ii)</td><td>Since there are 128 characters, 7 bits are needed/</td><td></td></tr>
<tr><td>(iii)</td><td>A check digit is an addition digit such that the sum of all the digits is even (or odd, depending on the system used). This allows single-digit errors to be detected.</td><td></td></tr>
<tr><td>(d)</td><td>There are more than 128 characters being used by the languages lin the world. If a text is written in multiple languages, there would need to be an encoding system that can accommodate much more than 128 characters.</td><td></td></tr>
<tr><td>2(a)</td><td>Full table:
<table>
<tr><td rowspan="3">Conditions</td><td>Software engineer</td><td>Y</td><td>Y</td><td>Y</td><td>Y</td><td>N</td><td>N</td><td>N</td><td>N</td></tr>
<tr><td>Contract based</td><td>Y</td><td>Y</td><td>N</td><td>N</td><td>Y</td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>Been with company more than 15 years</td><td>Y</td><td>N</td><td>Y</td><td>N</td><td>Y</td><td>N</td><td>Y</td><td>N</td></tr>
<tr><td rowspan="5">Actions</td><td>35 days of leave</td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>28 days of leave</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>25 days of leave</td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>X</td><td></td></tr>
<tr><td>21 days of leave</td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>18 days of leave</td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>X</td></tr>
</table>
Note: not necessary to create a separate condition for marketing personnel since if they are neither kind of engineer then they are in marketing. Likewise, you could do a table with Permanent Engineer and Marketing instead.</td><td></td></tr>
</table>

Simplified table (Someone can’t be both kinds of engineer simultaneously):

Conditions	Software engineer	Y	Y	Y	Y	N	N
	Contract based	Y	Y	N	N	-	-
	Been with company more than 15 years	Y	N	Y	N	Y	N
Actions	35 days of leave			X			
	28 days of leave	X			X		
	25 days of leave					X	
	21 days of leave		X				
	18 days of leave						X

2(b)



Accept variations in accessor and mutator methods

2(c)	Subclasses inherit attributes and methods from their parent classes. For example, contract engineers inherit the list of programming languages and the years of experience from the parent engineer class. The code is reused, so that changes can be easily made and debugging is easily done.	
2(d)	The salary calculation is different for each type of employee. However, using the same method name for all of them makes it easy to run code to compute the salary regardless of what type of employee they are.	
2(e)	The NRIC number and date of birth should be private attributes as those are personal data.	
2(f)	A 2-dimensional array where each row consists of two elements, the programming language and the number of years' experience with it.	
3(a)	1. Local DNS checks if URL is inside its jurisdiction, sends back IP address to the browser if it is. 2. If URL is not in its domain, it looks up URL in its cache to see if IP address is there. If it is, the IP address is sent back to the browser 3. If not, the DNS sends out a request to a root server, which provides an address for a DNS server with jurisdiction over the top-level domain, which can provide an address for a DNS server for the next level domain, and so on, until a server which can provide the IP address is found 4. The DNS adds this URL and IP address to its cache, and sends the IP address to the browser	
3(b)(i)	2^{32} (about 4.2 billion)	
3(b)(ii)	There are more than 4.2 billion unique devices.	
3(b)(iii)	There are $16^{32} = 2^{128}$ possible addresses, which is far more than the present number of devices, and will be for a long time to come.	
3(c)(i)	1. A message from one computer to another is broken up into individual (numbered) packets and each packet is sent separately. 2. The route for each packet is not determined in advance. It is sent from one computer to another in the same network, which then decides to send it to another computer which may be nearer to the destination, and so on, until it reaches the destination. Packets thus travel by different routes. 3. The destination computer receives the packets and assembles them in the correct order.	
3(c)(ii)	If computer is down, a connection between two computers is damaged, the packet can be rerouted to the destination by another route. Since the route is not predetermined, the path can be modified to work around changes (damage) in the network.	
3(d)	1. A malicious user sends many repeated requests to the server. 2. This overwhelms the server by using up available resources (memory and bandwidth) on the server, causing it to overload. 3. Legitimate users are not able to use the server, and their normal operations are disrupted.	
4(a)	Any two of: Worm, virus, trojan, ransomware, spyware	
4(b)	1. Downloading from FTP or webpage	

	2. Email attachment	
4(c)	1. Firewall 2. Antivirus software	
4(d)	1. Company's responsibility to keep software up to date against latest kind of viruses, and push updates out to users 2. Company should make sure software works as advertised, and not make false claims 3. Company should test software thoroughly before release, and resolve bugs that they have discovered	

5(a)

```
1  PROCEDURE InsertionSort(Arr: ARRAY OF INTEGER)
2     N ← LENGTH(Arr)
3
4     FOR i ← 2 TO N DO
5         key ← Arr[i]
6         j ← i - 1
7
9         WHILE j > 0 AND Arr[j] > key
10             Arr[j + 1] ← A[j]
11             j ← j - 1
12         ENDWHILE
13
13         Arr[j + 1] ← key
15    ENDFOR
16 ENDPROCEDURE
```

5(b)	O(N^2)	
5(c)	When the list is already sorted in ascending order. The condition for the while loop from line 9 to line 12 will not be fulfilled, hence the while loop will not be executed. The time complexity becomes O(N).	

5(d)

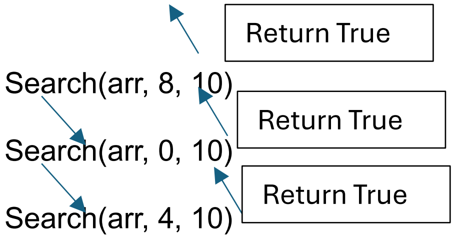
i	key	left	right	mid	A	j
2	1	1	1	1	3,1,6,5,4,2,7	1
					1,3,6,5,4,2,7	
3	6	1	2	1	1,3,6,5,4,2,7	2
					1,3,6,5,4,2,7	

4	5	1	3	2	1,3,6,5,4,2,7	3
					1,3,5,6,4,2,7	
5	4	1	4	2	1,3,5,6,4,2,7	4
					1,3,4,5,6,2,7	
6	2	1	5	3	1,3,4,5,6,2,7	5
					1,2,3,4,5,6,7	
7	7	1	6	3	1,2,3,4,5,6,7	6
					1,2,3,4,5,6,7	

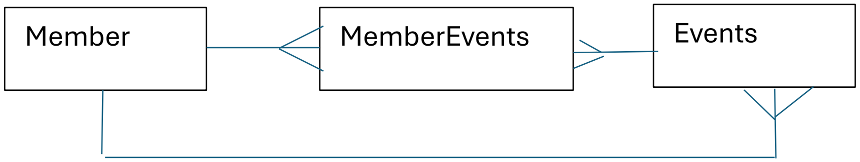
5(e)	Insertion sort uses linear search to find the correct position for the key to be inserted into. Binary insertion sort uses binary search to find the correct position. For a large list, this reduces the time taken to find the position. Binary insertion sort faster than insertion sort. *Note: For inserting the i-th element in its correct position in the sorted, finding the position (pos) will take O(log i) steps. However, to insert the element, we need to shift all the elements from pos to i-1. This will take i steps in the worst case (when we have to insert at the starting position).* *We make a total of N insertions — so, the worst-case time complexity of binary insertion sort is* ***O(N^2).*** *This occurs when the array is initially sorted in descending order.*	
5(f)	Line 12: IF A[mid] < key	
5(g)	O(N^2)	
5(h)	An algorithm is considered in-place if it sorts or processes data using only a small, constant amount of extra memory space beyond the input data itself. An algorithm is considered not in-place if it requires additional memory that scales with the size of the input data.	

5(i)	Advantage: Minimal additional memory usage, which can be important in memory-constrained environments. Disadvantage: The original data is changed, which might not be desirable if the original data needs to be preserved.	
6(a)	Any 3 of the below points and a conclusion Hash table has constant search/insert/delete time O(1), while BST has O(lgn), so hash table search is faster Collisions might occur in hash tables that increases the search time, and it could be O(n) in the worst case. BST could have O(n) search time if the tree is unbalanced, while collisions are rare when the hash function is good. BST can get list of sorted items by doing in-order traversal, but not for hash table BST is more memory efficient as it does not require more memory than necessary but hash tables require a lot more memory than required to prevent collisions.	
6(b)	• has a time complexity of O(1) • returns a unique output when provided a unique input uses the entire input to determine the address	
6(c)	10, 40, 30, 90, 80, 50	

6(d)	Function search(arr, Root, target)
	    IF Root == None THEN // 1
	        RETURN False
	    ELSE IF target == arr[Root][0] THEN // 1
	        RETURN True
	    ELSE IF target < arr[Root][0] THEN // 1
	        RETURN search(arr, arr[Root][1], target)
	    ELSE IF target > arr[Root][0] THEN // 1
	        RETURN search(arr, arr[Root][2], target)	

6(e)	 Return True Search(arr, 8, 10) Return True Search(arr, 0, 10) Return True Search(arr, 4, 10)	
6(f)	Iterative algorithm has constant space requirement regardless of how long the list is. The amount of memory space required for recursive algorithm is depending on the number of recursive calls made. More recursive calls (which usually occurs when list is longer) leads to greater memory use.	

7(a)	Member(MemberID, MemberName, Email, phone, points)
	
	MemberEvents(MemberID*, EventName*)
	
	Events(EventName, distance, location, number1*, number2*, number3*, organizerID*)
	
	Alternative answer
	
	Member(MemberID, MemberName, Email, phone, points)
	Events(EventName, distance, location, organizerID*)
	MemberEvents(MemberID*, EventName*, points)	

7(b)	MemberID in MemberEvents ensures each event is joined by a valid member. EventName in MemberEvents ensures each member joins a valid event.	
7(c)	 	

7(d)	Ensure all non-key attributes are fully functional dependent on the entire primary key in each table; hence the tables are in 2NF. For the 2NF tables, ensure that there is no transitive dependency for all the non-primary key attributes in the tables.	
7(e)	SELECT Member.MemberName FROM Member INNER JOIN MemberEvents ON Member.MemberID = MemberEvents.MemberID WHERE EventName = “Charlestown Marathon 2002” AND Member.points < 20	