JURONG PIONEER JUNIOR COLLEGE

2022 JC2 Preliminary Examination

COMPUTING
Higher 2

9569/01

19 September 2022

Paper 1 (Written)

3 hours

Answer & Mark Scheme

1	**(a) (i) Data Integrity** • Data integrity is the **accuracy, and completeness maintained over time and across formats [1m]**. Preserving the data integrity is a constant process.	Total 1 mark
	(ii) Two threats: • **Human error**: For instance, accidentally deleting a row of data in a spreadsheet • **Inconsistencies across format**: For instance, a set of data in Microsoft Excel that relies on cell referencing may not be accurate in a different format that doesn't allow those cells to be referenced • **Collection error**: For instance, data collected is inaccurate or lacking information, creating an incomplete picture of the subject	Total 2 marks Choose any 2 points (without elaboration)

	• **External or internal cybersecurity or privacy breaches**: For instance, someone hacks into your company's database with the intent to damage or steal information, or an internal employee damages data with malicious intent	
	(iii) **Data integrity** refers to the **completeness, consistency, accuracy and the validity of the data [1m]**. ie. when recorded, it is recorded exactly as the user intends, and when retrieved, it is in the exact same state that it was recorded. **Data security** is the practice of **preventing data from being accessed, altered, disclosed, or damaged without authorisation [1m]**. The term also covers the methods used to do this. These include	Total 2 marks 1 mark for each point adequately mentioned.
	(iv) Some data may not be being used very often but it may still be useful or needed in the future. In this case data can be archived. **Archived data is copied to a suitable storage medium (perhaps DVDs or magnetic tape) then it is stored safely and securely.** The original data is then deleted from the computer system. **This is done to free up resources for new data**.	Total 1 mark
	(b) Data validation is the process **comparing** information with a **set of rules** to ensure it is **sensible and reasonable [1m]** enough before it is stored in the system.	Total 2 marks

Data integrity ensures correctness or accuracy of the data, while data validation only checks if data is within an **acceptable range/expected range, hence it is unable to check for the correctness or accuracy of the data [1m]**.

(c) Indiscriminate or negligent handling of NRIC numbers increases the risk of unintended disclosure. NRIC/FIN may be obtained and used for **illegal activities such as identity theft and fraud as they can be used to unlock large amounts of personal information relating to the individual [1m].** The NRIC/FIN is also **a permanent and irreplaceable identifier specific to an individual [1m].**

Total 2 marks

(d) Organisations can consider to:

- **develop**, **implement and regularly review data policies and practices** to keep them abreast and updated. **[1m]**
- employ authorise **third party data specialist professinals** to help them **store and management data**. **[1m]**

Total 2 marks

(e) **(i)**

Birthday in *ddmm*				**NRIC 3 rightmost integers**			**Check Digit**	
8	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**← digit position**
0	1	0	2	4	3	2		**8-digit** `TenantID`

Total 2 marks

	Product sum = (0*8) + (1*7) + (0*6) + (2*5) + (4*4) + (3*3) + (2*2) = 0 + 7 + 0 + 10 + 16 + 9 + 4 = 46 C = [11 - (46 MOD 11)] MOD 11 = [11 – 2] MOD 11= **9** Check digit of tenant A is **9**	1m – attempt to calculate sum of the product 1m – Obtain check digit = 9
	(ii) **Birthday in *ddmm*** | **NRIC 3 rightmost integers** | **Check Digit** **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **← digit position** 1 | 7 | 1 | 2 | 8 | 9 | 6 | | **8-digit** `TenantID` Product sum = (1*8) + (7*7) + (1*6) + (2*5) + (8*4) + (9*3) + (6*2) = 8 + 49 + 6 + 10 + 32 + 27 + 12 = 144 C = [11 - (144 MOD 11)] MOD 11 = [11 – 1] MOD 11 = 10 Check digit = X (using X to represent 10 in the 8 digit number string code) Therefore, the check digit the `tenantID` of `B = 1712896`**X**	Total 2 marks 1m – Obtain checkdigit = 10 1m – present check digit represented as by a symbol stated.
	(f) **Birthday in *ddmm*** | **NRIC 3 rightmost integers** | **Check Digit** **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **← digit position** 0 | 9 | 0 | 8 | 9 | 9 | 5 | **3** | **8-digit** `TenantID` Product sum = (0*8) + (9*7) + (0*6) + (8*5) + (9*4) + (9*3) + (5*2) + (3*1)	Total 2 marks

	= 0 + 63 + 0 + 40 + 36 + 27 + 10 + 3 = 179 Since 179 MOD 11 <> 0 Therefore `TenantID` of value '`09089953`' is not a valid 8 digit number string	1m – obtain product sum = 179 1m – state 179 MOD 11 <> 0, and conclude `tenantID` not valid.
	(g) **(i)** A primary key **ensures unique row identification [1m]**. This results in **faster sorting, searching, and querying** operations **[1m]**. **(ii)** A foreign key **creates a link between two tables [1m]**. It **maintains referential integrity [1m]** between the referencing column and the referenced column.	Total 2 marks Total 2 marks

	(h)
	(i)
	
	`Tenant(tenantID,Name,sex,DOB,email,Contact_No)`
	
	`Apartment(apartmentID,level,unit,occupied)`
		Total 2 marks
		1m – correct PK underlined and seen in both tables.
		1m – 2 other sensible fields

	(ii) `RentalContract(tenantID, apartmentID, startDate, endDate, bookingDate)` (by selecting as the composite keys will not be sufficiently unique enough to uniquely identify the records as they grow in size.)	Total 1 mark 1m – correct composite keys underlined.
	(i) ER diagram	Total 3 marks 1m – all entities correct

		1m – r/s between TENANT and RENTALCONTRACT 1m – r/s between APARTMENT and RENTALCONTRACT
	(j) The foreign key `TenantID` in table `RentalContract` references to primary key `TenantID` of the of the table/entity `Tenant` **[1m]** The foreign key `apartmentID` in table `RentalContract` references to primary key `apartmentID` of the of the table/entity `apartment`. **[1m]**	Total 2 marks
	(k) Data inconsistency happens when the **same data element/object** stored in **multiple files/locations** within a database is found to contain **different information of itself [1m]**. Data redundancy occurs when the **same data element/object exists in multiple locations/files [1m]** within the database.	Total 2 marks
	(l) Having a large number of the same data element stored in multiple locations would **increase the likelihood of partial update [1m]** where the data element gets **updated in one/some location(s) but not the others [1m]** which is a consequence of inconsistent data.	Total 2 marks

2	**(a)**	**(i)** Array stores all its data elements continguously in order not to create space wasteages. However maintaining **data elements stored continguously requires high memory resources that is at the expense of performance**. Inserting an item into a non-empty array at index **location `i` in an array would first require moving all the existing data items from array locations `i+1` onwards to the last data item in the array one location to the right [1m]**. Similary **after deleting a data item in location `i` of the array would require every data items from locations `i+1` onwards to the last item forward by one location [1m]** in order to ensure data items remain stored continguously. **(ii)** For insertion and deletion of data item in a linked list, only the **affected pointer values need to be updated [1m]**.	Total 3 marks Total 1 mark
	(b)	Inserting a node to the front of a linked list: • Creation and initialisation of a new node that contains the `data_pointer = newItem`, and `next_pointer = NULL` • IF `Start` pointer is `NULL` THEN set `start` to point to new node/address of new node. • Else, store the value of `start` pointer into a temporary variable • `start` pointer to point to the new node. • `next` pointer of the new node to store the value of the temporary variable (`next` to point to the node that was previously the start node)	Total 3 marks 1m – initialise node with data and next pointer = NULL and if Linked List empty, set start to point to new node

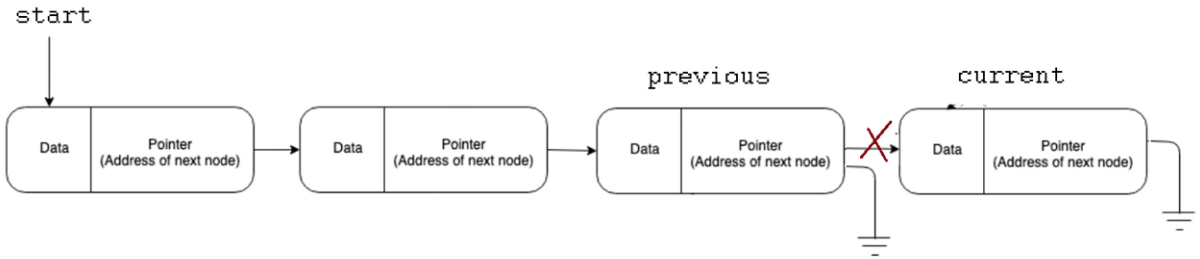
(c) Deleting a node from the rear of non-empty linked list:

If linked list contains **1 node** before deletion:

- Set `start` to `NULL` **[1m]**
- Deallocate and return the current node back to the main memory

If linked list contains **more than 1 node** before deletion:

- Using 2 pointers `previous` and `current` to traverse in tandem from the first node of the linked list until current reaches the last node (ie. next pointer of node points to null). **[1m]**
- Set next pointer of previous node to NULL. **[1m]**
- Deallocate and return the current node back to the main memory



1m – if Linked List not empty, set start into temp var,

1m - set start to new node, and set next_pointer of new node to temp var.

Total 3 marks

1m – to consider if linked list contains 1 node before deletion and set start to null

1m – for traversing `L` from start with `prev` and `cur` until `cur` reaches the last node.

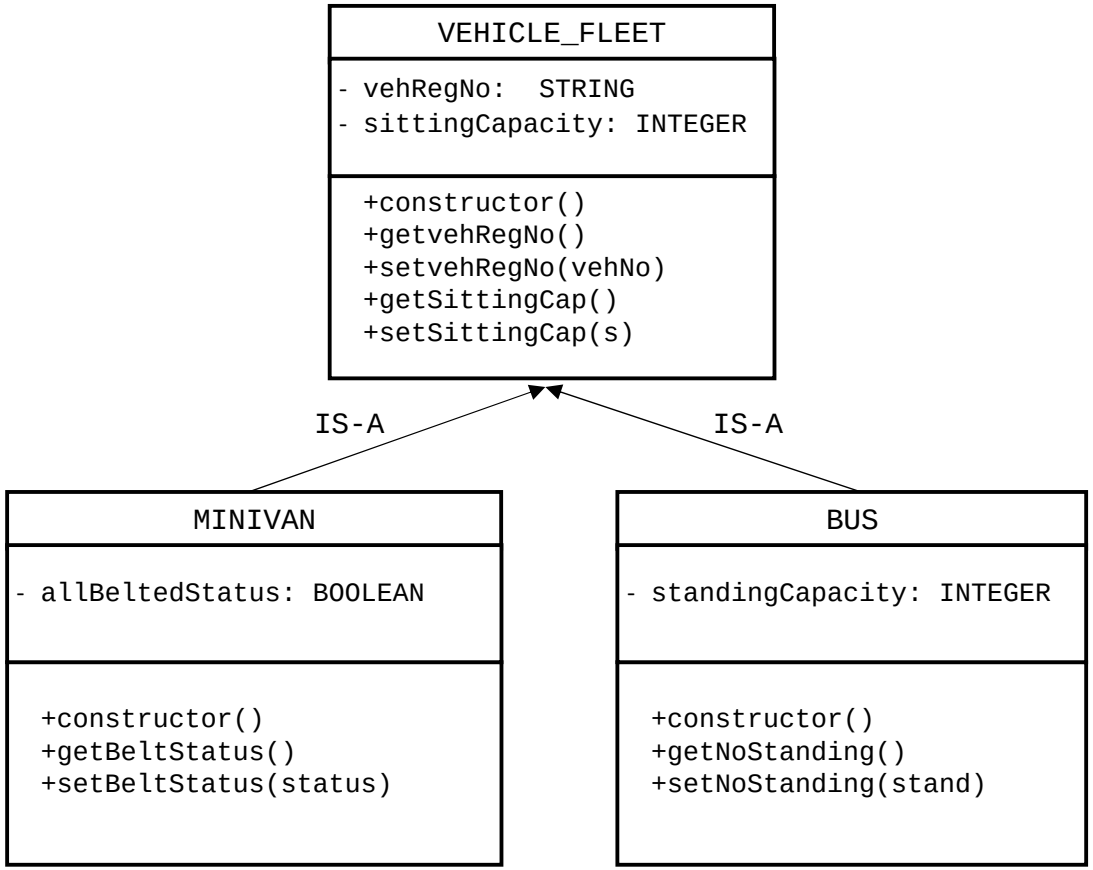
1m for update `prev.next = null`

	(d) **Slower search/traversal time** as it **does not allow direct access to node** or traversal needs to **always start from the first node**. (SR: No marks if student answers "does not allow direct access or trversal starts from first node as they do not answer the question directly) **(e)** ``` FUNCTION find(k) returns BOOLEAN current = start – 1m WHILE current <> NULL – 1m IF current.data = k THEN – 1m RETURN True ENDIF current = current.next END WHILE RETURN False ENDFUNCTION ``` **(f)** **(i)** `Create(Q)` **(ii)** `Insert(Q, data, Length(Q))` **(iii)** ``` IF isEmpty(Q) = True THEN -1m PRINT('Empty Queue') ENDIF Read(Q,1) -1m Delete(Q,1) -1m ```	Total 1 mark Total 3 marks 1m – initialise current to start 1m – while loop to check if the end is reached. 1m – if statement to check if search key is found. Total 1 mark Total 1 mark Total 3 marks
3	**(a)** 1. First chooses the first item in the array as the **pivot**. 2. **Partitions** by **<u>moving</u>** the items in the data set **about the pivot** items to smaller than pivot to the left and larger to the right.	Total 4 marks 1m for 1 point

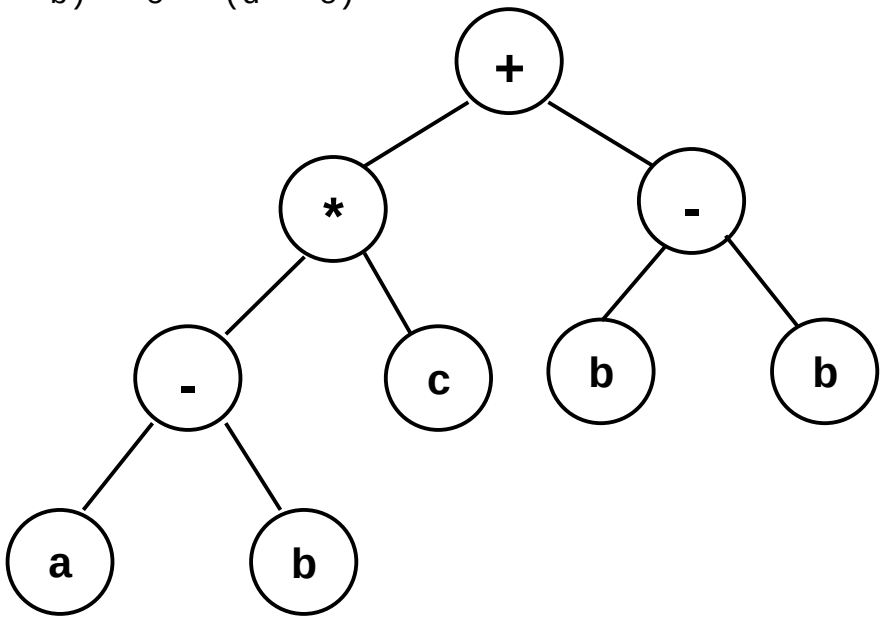
<table>
<tr><td></td><td>3. This would result in the pivot be in a position that splits/partitions the data set into 2 parts (also known as the spilt point).
4. Recursively performs the tasks 1, 2 and 3 on each sub-array partitioned u the spilt point until data until 1 item remains, and the data set will be sorted.</td><td></td></tr>
<tr><td></td><td>(b) Best case time complexity = $O(n\log_2 n)$</td><td>Total 1 mark</td></tr>
<tr><td></td><td>(c) The worst case scenario happens when the chose pivot in each of the call ends up spilting the array of n elements [1m] into 2 sub-arrays of size 1 and n-1 [1m].</td><td>Total 2 marks</td></tr>
<tr><td>4</td><td>(a) The three programming constructs are:
Sequence, Selection, Iteration (Repetition)
(Do not accept if students answers IF-THEN-ELSE for selection, or LOOPS for iteration without mentioning the actual keywords).</td><td>Total 2 marks
3 correct 2 marks
2 correct 1 mark
0 mark otherwise</td></tr>
<tr><td></td><td>(b) Sequence and Selection</td><td>Total 2 marks
Each word 1 mark</td></tr>
<tr><td></td><td>(c) Use meaningful names for identifiers.</td><td>Total 1 mark (exactly)</td></tr>
<tr><td></td><td>(d) Line 5</td><td>Total 1 mark</td></tr>
<tr><td></td><td>(e)
<table>
<tr><th>n</th><th>Recursive call</th><th>Print</th></tr>
<tr><td>36</td><td>X(36)</td><td></td></tr>
</table>
</td><td></td></tr>
</table>

		18	X(18)			
		4	X(4)			
		2	X(2)			
		1	X(1)	1		
				0		
				0		
				1		
				0		
				0		
	(f) X is a procedure that converts prints a denary number to its binary equvalent					Total 1 mark
5	**(a)** Object is an instance of a class [1m]. Class is a blueprint or template from which objects are created [1m].					Total 2 marks

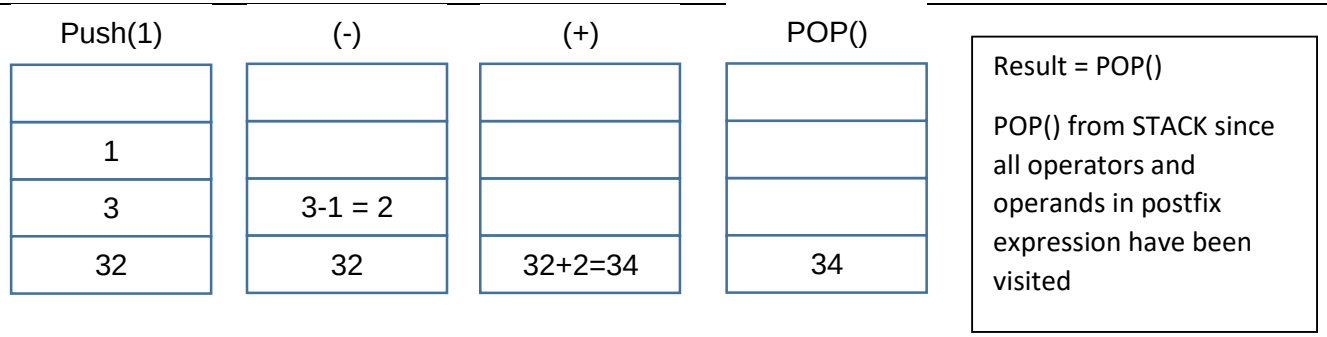
(b)



Total 4 marks

<table>
<tr>
<td>6</td>
<td>(a) <code>(a – b) * c + (d – e)</code>
 
</td>
<td>Total 3 marks
1m – correct root</td>
</tr>
<tr>
<td></td>
<td>(b) <code>62-8*31-+</code></td>
<td>Total 1 mark</td>
</tr>
<tr>
<td></td>
<td>(c) Traverse <code>62-8*31-+</code> item by item from the left to the right

<table>
<tr><th>Push (6)</th><th>Push (2)</th><th>(-)</th><th>Push(8)</th><th>(*)</th><th>Push(3)</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>2</td><td></td><td>8</td><td></td><td>3</td></tr>
<tr><td>6</td><td>6</td><td>6-2 = 4</td><td>4</td><td>4*8=32</td><td>32</td></tr>
</table></td>
<td>Total 3 marks
1m observe push operand into stack
1m observe pop twice and evaluate operator and push evaluated result back into stack if end of postfix not reached.</td>
</tr>
</table>

		1m observe pop final result out from stack when the postfix has reached the end.
7.	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.	Total 5 marks 1 mark for 1 step except step 6.
8.	**(a)** The goal of network security is to achieve the following: **Confidentiality** - Ensure that private data that is **disclosed or made available only to authorised persons or organisations** and be **protected from any unauthorised access**. **Integrity** - to make sure that **data is accurate and consistent**, and not changed by unauthorised personnel.	Total 3 marks 1m for each point with correct reason.

	Availability - data, network resources/services are continuously available to the authorised users, whenever they require it.	
	(b) In P2P network, the **absence of a centralised server** would mean that **any device connected** to the requesting device **can share a fragment** of the resource to the requesting device. Therefore, there is **no way for any parties to control what content is being transmitted** from the senders and the receiver, which could carry risks and vulnerabilities relating to data integrity, **viruses, spyware, adware, and unwanted files**.	Total 2 marks 1m – no centralised server, any device can be send. 1m – no centralised server, unable to have regulations on the data sent resulting in viruses, spyware, adware etc sent to the receiver.
	(c) A firewall is : **1.** a **barrier between a trusted internal network** and **untrusted external network**, such as the Internet **[1m]**. **2.** can be a software program or a hardware device, or a combination of both that **monitors and controls incoming and outgoing network traffic based on predetermined security rules [1m].** **3.** Serves as a **barrier between a device's internal network and the incoming traffic from external sources (such as the internet)** with an intention to **block malicious traffic like viruses and hackers**, and they **allow incoming traffic sent as a response to requests from internal hosts [1m]**.	Total 3 marks