

2024 JC2 Prelim Solution

No	Solution	Marks																																																																																																																
1	<table><tr><th>Condition</th><th>C1</th><th>C2</th><th>C3</th><th>C4</th><th>C5</th><th>C6</th><th>C7</th><th>C8</th></tr><tr><td>Short Hair</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>Wear Spectacle</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>Female</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>Outcome</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Soup served</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td><td></td></tr><tr><td>Gold plated steak</td><td>X</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Chef cut steak for you</td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>X</td><td></td></tr></table> <table><tr><th>Condition</th><th>C1/2</th><th>C3/4</th><th>C5/7</th><th>C6/8</th></tr><tr><td>Short Hair</td><td>Y</td><td>Y</td><td>N</td><td>N</td></tr><tr><td>Wear Spectacle</td><td>Y</td><td>N</td><td>-</td><td>-</td></tr><tr><td>Female</td><td>-</td><td>-</td><td>Y</td><td>N</td></tr><tr><td>Outcome</td><td></td><td></td><td></td><td></td></tr><tr><td>Soup served</td><td>X</td><td>X</td><td></td><td></td></tr><tr><td>Gold plated steak</td><td>X</td><td></td><td></td><td></td></tr><tr><td>Chef cut steak for you</td><td></td><td></td><td>X</td><td></td></tr></table>	Condition	C1	C2	C3	C4	C5	C6	C7	C8	Short Hair	Y	Y	Y	Y	N	N	N	N	Wear Spectacle	Y	Y	N	N	Y	Y	N	N	Female	Y	N	Y	N	Y	N	Y	N	Outcome									Soup served	X	X	X	X					Gold plated steak	X	X							Chef cut steak for you					X		X		Condition	C1/2	C3/4	C5/7	C6/8	Short Hair	Y	Y	N	N	Wear Spectacle	Y	N	-	-	Female	-	-	Y	N	Outcome					Soup served	X	X			Gold plated steak	X				Chef cut steak for you			X		6 One mistake -1 mark
Condition	C1	C2	C3	C4	C5	C6	C7	C8																																																																																																										
Short Hair	Y	Y	Y	Y	N	N	N	N																																																																																																										
Wear Spectacle	Y	Y	N	N	Y	Y	N	N																																																																																																										
Female	Y	N	Y	N	Y	N	Y	N																																																																																																										
Outcome																																																																																																																		
Soup served	X	X	X	X																																																																																																														
Gold plated steak	X	X																																																																																																																
Chef cut steak for you					X		X																																																																																																											
Condition	C1/2	C3/4	C5/7	C6/8																																																																																																														
Short Hair	Y	Y	N	N																																																																																																														
Wear Spectacle	Y	N	-	-																																																																																																														
Female	-	-	Y	N																																																																																																														
Outcome																																																																																																																		
Soup served	X	X																																																																																																																
Gold plated steak	X																																																																																																																	
Chef cut steak for you			X																																																																																																															
2a	Start stack ← Stack() current ← root Current <> NULL OR NOT stack.isEmpty()? END current <> NULL? stack.push(current) current ← current.left current ← stack.pop() OUTPUT current.data current ← current.right *root is the root of node of the BST	5																																																																																																																
2b	Output BST nodes data In order	1																																																																																																																
2c	07	1																																																																																																																

2d	<table><tr><th>line</th><th>current</th><th>stack</th><th>OUTPUT</th></tr><tr><td>03</td><td>4</td><td>empty</td><td></td></tr><tr><td>07</td><td>4</td><td>[4</td><td></td></tr><tr><td>08</td><td>3</td><td></td><td></td></tr><tr><td>07</td><td>3</td><td>[4,3</td><td></td></tr><tr><td>08</td><td>None</td><td></td><td></td></tr><tr><td>10</td><td>3</td><td>[4</td><td></td></tr><tr><td>11</td><td></td><td></td><td>3</td></tr><tr><td>12</td><td>None</td><td></td><td></td></tr><tr><td>10</td><td>4</td><td>empty</td><td></td></tr><tr><td>11</td><td></td><td></td><td>4</td></tr><tr><td>12</td><td>9</td><td></td><td></td></tr><tr><td>07</td><td></td><td>[9</td><td></td></tr><tr><td>08</td><td>7</td><td></td><td></td></tr><tr><td>07</td><td></td><td>[9, 7</td><td></td></tr><tr><td>08</td><td>None</td><td></td><td></td></tr><tr><td>10</td><td>7</td><td>[9</td><td></td></tr><tr><td>11</td><td></td><td></td><td>7</td></tr><tr><td>12</td><td>None</td><td></td><td></td></tr><tr><td>10</td><td>9</td><td>empty</td><td></td></tr><tr><td>11</td><td></td><td></td><td>9</td></tr><tr><td>12</td><td>None</td><td></td><td></td></tr></table>	line	current	stack	OUTPUT	03	4	empty		07	4	[4		08	3			07	3	[4,3		08	None			10	3	[4		11			3	12	None			10	4	empty		11			4	12	9			07		[9		08	7			07		[9, 7		08	None			10	7	[9		11			7	12	None			10	9	empty		11			9	12	None			4
line	current	stack	OUTPUT																																																																																							
03	4	empty																																																																																								
07	4	[4																																																																																								
08	3																																																																																									
07	3	[4,3																																																																																								
08	None																																																																																									
10	3	[4																																																																																								
11			3																																																																																							
12	None																																																																																									
10	4	empty																																																																																								
11			4																																																																																							
12	9																																																																																									
07		[9																																																																																								
08	7																																																																																									
07		[9, 7																																																																																								
08	None																																																																																									
10	7	[9																																																																																								
11			7																																																																																							
12	None																																																																																									
10	9	empty																																																																																								
11			9																																																																																							
12	None																																																																																									
2e	4 3 9 7 20 6 15 25	2																																																																																								
2f	Pre-order: 4, 3, 9, 7, 6, 20, 15, 25 Post-order: 3, 6, 7, 15, 25, 20, 9, 4	2																																																																																								
3a	foo(4,2) # return 2 + (-6) = -4 foo(3,0) # return 0 + (-6) = -6 foo(2,-2) # return -2 + (-4) = -6 foo(1,-4) # return (-4) + 0 = -4 foo(0,-6) # return 0 Corrected version foo(4,2) # return 2 + (-6) = -4 2 + foo(3,0) # return 0 + (-6) = -6 0 + foo(2,-2) # return -2 + (-4) = -6 -2 + foo(1,-4) # return (-4) + 0 = -4 -4 + foo(0,-6) # return 0	3																																																																																								
3b	FUNCTION foo(n, m) RETURNS INTEGER result ← 0 WHILE n > 0 DO result ← result + m n ← n - 1 m ← m - 2 END WHILE RETURN result END FUNCTION	3																																																																																								
4a	Network delays, jitter or different packet routes may cause later times to arrive before earlier ones. [different client locations accepted]	1																																																																																								
4b	Array supports binary search O(log n) linked list search is sequential O(n).	2																																																																																								

4c	Insertion into the middle of a sorted array is $O(n)$ (shifting required), but in a linked list insertion (after locating the position) is $O(1)$. Not accepting sorted array can only accommodate fixed number of record as the number of expected record is known in advance.	1
4d	Since finishing times usually arrive in increasing order, new records are appended at the end of the array. Even if the finishing time records may arrive out of order, the insertion took place with minimal shifts since the finishing time is generally of a large value. Appending takes $O(1)$ time, so in practice insertion is much faster than the worst-case $O(n)$.	2
4e	Use a digital signature for each message.	1
4f	The client computes a hash digest of the finishing time message. The client encrypts the hash with its private key to form the digital signature. The client sends both the message and the signature to the server. The server computes its own hash of the received message. The server decrypts the signature using the client's public key to recover the original digest. The server compares the two digests: if they match $\rightarrow$ the message is authentic and unaltered; if not $\rightarrow$ it has been tampered with or forged.	6
5a	1. Collisions Different keys may hash to the same index (bucket). When collisions occur, elements must be stored together (e.g., chaining or probing). This can degrade search time from $O(1)$ to $O(n)$ in the worst case if many keys collide. 2. Poor hash function If the hash function does not distribute keys uniformly, certain indices become overloaded. This leads to long chains/clusters at a few positions, making searches slower. 3. High load factor / table too full As the number of stored elements approaches or exceeds the table's capacity, collisions increase. Without resizing and rehashing, search time will degrade significantly beyond $O(1)$.	3
5b	The method is used to handle cases when two or more keys map to the same hash value. (Cannot be just specific to linear probe)	1
5c	i) Closed addressing ii) Open addressing	1
6a	Data validation: an automated process that checks if the data entered is reasonable, sensible, and in the correct format (e.g., ensuring a date field contains a valid date). Data verification: a process to confirm that the data entered is accurate and matches the user's intention (e.g., asking a user to enter the same data twice to reduce typing errors).	2
6b	Check that the date entered is a valid calendar date and not a future date .	1
6c	To ensure that the user typed the email address correctly and did not make a typographical error. Its another purpose is to verify that the user's intended email address	2
6d	Use Unicode (e.g., UTF-8) encoding instead of ASCII so the system supports characters from multiple languages.	1
6e	Obtain the user's consent and state the purpose of collecting their personal data. Implement reasonable security measures to protect personal data from unauthorized access or disclosure.	2

7ai	Application layer The engineer checks: • whether the web server application (HTTP/HTTPS) is running • whether the DNS resolution works correctly • if the application itself has crashed or configuration is wrong.	2
7aii	Network Access layer The engineer checks: • Faulty Ethernet cable • Wi-Fi not associated • ARP resolution failure	2
7aiii	Transport layer The engineer checks: • Correct port number used • Firewall rules • Session setup process	2
7aiv	Internet layer The engineer checks: • IP address • subnet mask, • default gateway, • routing configuration	2
7b	TCP is connection-oriented, UDP is connectionless TCP provides reliable delivery through knowledgements, retransmissions, and error checking ensure data arrives in order and without lost. UDP provides best-effort delivery .	2
7c	opened a phishing email attachment without verifying its authenticity	1
7d	ransomware	1
7e	Recovery measure: Maintain regular backups of critical data so it can be restored if encrypted.	1
7f	Preventive measure: Keep systems patched and updated , and use email filtering and user awareness training to reduce phishing risk.	1
8	C- 1 mark for 4 classes I - 1 mark for correct inheritance shown (hollow arrow heads) Appliance is not a subclass of power unit A – 1 mark for all correct attribute with data type and getters/setters M - 1 mark for identification of appropriate methods C – 1 mark for constructor P - 1 mark for polymorphism – 2 connect in subclass PowerUnit - type: str - power rating: int - brand: str + PowerUnit (type: str, power_rating: int, brand: str) + getters/setters of all attributes Appliance - power_unit: PowerUnit - colour: str - weight: float - is_on: Boolean + Appliance (power_unit: PowerUnit, colour: str, weight: float, is_on: Boolean) + getters/setters of all attributes + turn_on() + turn_off() + connect_home_network()	6

	<pre> classDiagram class WashingMachine { - drum_size: int + WashingMachine (power_unit: PowerUnit, colour: str, weight: float, is_on: Boolean, drum_size: int) + getters/setters of all additional attributes + connect_home_network() } class Refrigerator { - cooling_cap: int + Refrigerator (power_unit: PowerUnit, colour: str, weight: float, is_on: Boolean, cooling_cap: int) + getters/setters of all additional attributes + connect_home_network() } WashingMachine < -- Refrigerator </pre>	
8b	Polymorphism is the ability of different classes to define the same method name but provide different implementations. Circle connect_home_network() in the sub classes	2
8c	Encapsulation is the practice of restricting direct access to an object's internal data and only allowing it to be accessed or modified through controlled methods (getters/setters). Example: In the Appliance class, the attribute powerUnit could be kept private , and the class could provide methods getPowerUnit() and setPowerUnit() to safely access or modify it.	2
9a	Social issue – NRIC number is revealed; sensitive information may be accessible by others that may be detrimental to the resident; No privacy and personal data protection. Financial issue – admin may missed out keying in status and result payment issues.	1 1
9b	i) repeated booking grouping for each court; ii) fields are not atomic iii) fields value are not consistent 'Paid' and " for Unpaid	1 1
9c	Data redundancy is present. The NRIC information for a resident is repeated for each of the booking.	2
9d	Normalisation is to reduce redundancy and improve readability/management of data	2
9e	The database is not in 3rd normal form as resident_id will not be directly dependent on the Primary Keys. resident_id is directly dependent on booking_id which cannot be the PK by itself in this table	2
9f	Resident (resident_id, name, nric_last_4) Booking (booking_id, resident_id, payment) Booking_Schedule (booking_id, date, court, time)	2 3 4
9g	Resident >-- Booking --< Booking_Schedule	4