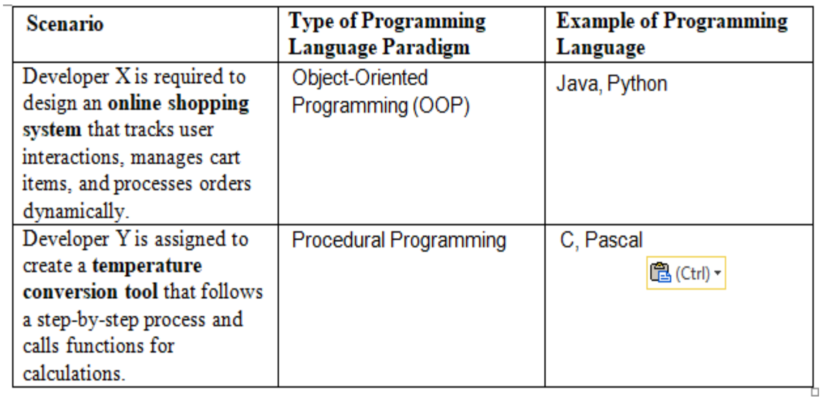
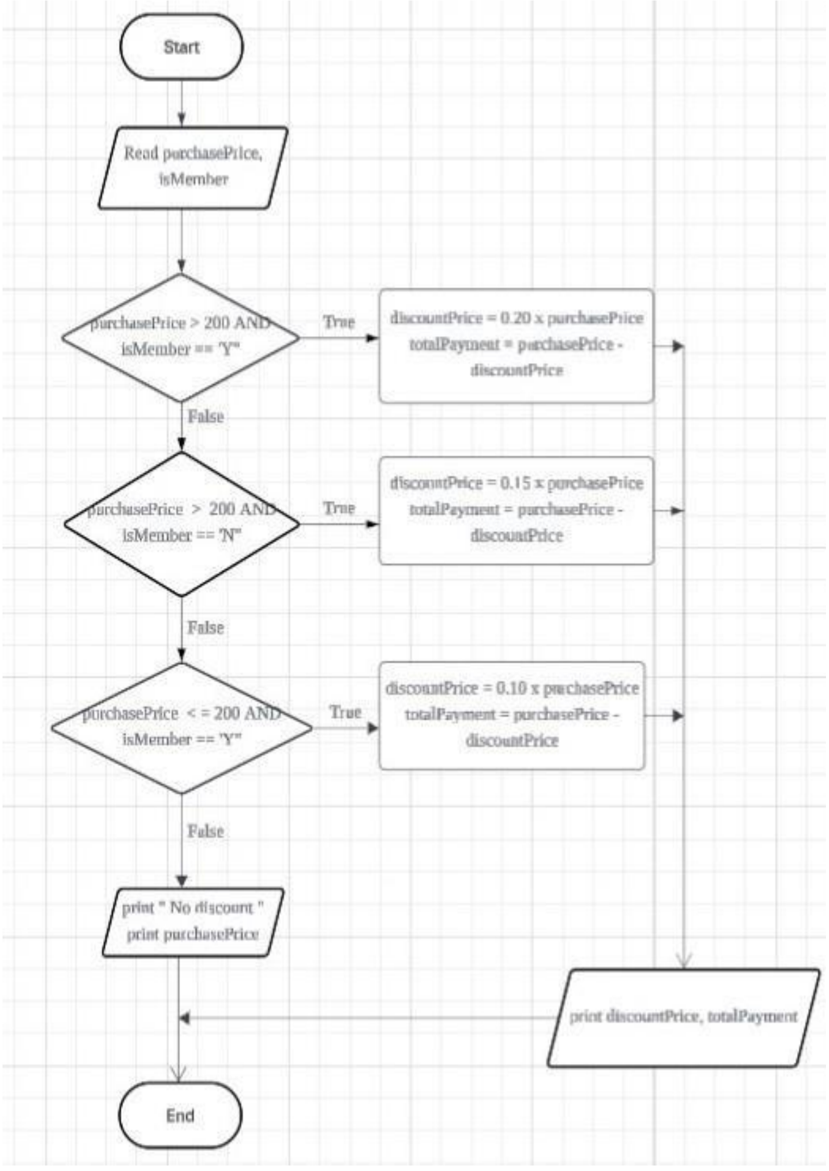
<table>
<tr><th>No.</th><th>Answers</th><th>Marks Allocation</th></tr>
<tr><td>1 a)</td><td>
<table>
<tr><th>Aspect</th><th>Procedural Programming</th><th>Object-Oriented Programming</th></tr>
<tr><td><b>Code Reusability</b></td><td>Achieved through pure functions and higher-order functions, promoting modularity. Limited to function reuse.</td><td>Achieved through inheritance and polymorphism, allowing code reuse across multiple classes.</td></tr>
</table>
</td><td>Each column 1 mark</td></tr>
<tr><td>1 b)</td><td>

<table>
<tr><th>Scenario</th><th>Type of Programming Language Paradigm</th><th>Example of Programming Language</th></tr>
<tr><td>Developer X is required to design an <b>online shopping system</b> that tracks user interactions, manages cart items, and processes orders dynamically.</td><td>Object-Oriented Programming (OOP)</td><td>Java, Python</td></tr>
<tr><td>Developer Y is assigned to create a <b>temperature conversion tool</b> that follows a step-by-step process and calls functions for calculations.</td><td>Procedural Programming</td><td>C, Pascal</td></tr>
</table>
</td><td>Each column 1 mark</td></tr>
<tr><td>2 a)</td><td>Input - number of hours the air conditioner is used
Process - calculate the total electricity cost for the day based on the number of hours the air conditioner is used
Output - total electricity cost for the day</td><td>1m
2 m
1m</td></tr>
<tr><td>2 b)</td><td>Input - gas pressure reading
Process - repeat for every hour
- <b>determine whether the system will trigger an alert</b> OR <b>display status message based on gas pressure reading</b>
Output - trigger alert if pressure exceeds the limit or status message if the pressure is below safe limit
OR
Output - message “Alert!” or “Normal”</td><td>1m
1m
1 + 1m
1m
1m
OR
2m</td></tr>
</table>

3 a)	START	
	TotalWater = 0	1m
	Read CropID, numHectare	1m
	IF CropID == “CC”	1m
	Display message “Corn”	
	TotalWater = numHectare * 800	
	ELSE IF CropID == “RC”	1m
	Display message Rice”	
	TotalWater = numHectare * 1500	
	ELSE IF CropID == “WC”	1m
	Display message “Wheat”	
	TotalWater = numHectare * 600	
	ELSE	
	DISPLAY message “Invalid crop ID”	1m
	ENDIF	
	END	
3b)	 Start Read purchasePrice, isMember purchasePrice > 200 AND isMember == "Y" True discountPrice = 0.20 x purchasePrice totalPayment = purchasePrice - discountPrice False purchasePrice > 200 AND isMember == "N" True discountPrice = 0.15 x purchasePrice totalPayment = purchasePrice - discountPrice False purchasePrice < = 200 AND isMember == "Y" True discountPrice = 0.10 x prchasePrice totalPayment = purchasePrice - discountPrice False print " No discount " print purchasePrice print discountPrice, totalPayment End	Read input Read purchasePrice - J 1 Read isMember - J 1 Condition if AND isMember == “Y” - J1 Condition if AND isMember == “N” - J1 Condition if AND isMember== “Y” - J1 Process discountPrice= 0.20 x purchasePrice - J1 totalPayment= purchasePrice-discountPrice - J1 Process discountPrice= 0.15 x purchasePrice - J1 totalPayment= purchasePrice-discountPrice - J1 Process discountPrice= 0.10 x purchasePrice - J1 totalPayment= purchasePrice-discountPrice - J1

		print discountPrice - J 0.5 print totalPayment - J 0.5 print “ No discount “ - J 0.5 print purchasePrice - J 0.5 True / False (3 pair) - J1 ***Accept any logical flow*** Deduct missing pair J0.5 illogical flow J1
4 a) i)	5 2.5	1m 1m

4 a) ii)	i = 2
	while(i < listStudents - 1)
	print(i)
	i = i + 2
		0.5m
		0.5m
		0.5m
		0.5m

4b (i)	$y = 3x^5 + 4x^2 - 9$ y= (3* (Math.pow(x,5)) +(4*(Math.pow(x,2))) -9 ;	3* (Math.pow(x,5) 1m 4*(Math.pow(x,2) 1m
4b(ii)	$y = \frac{x^{2} -16x +64}{2x -16}$ y= ((x*x) - (16*x) + 64)/(2*x -16);	y= 0.5 m (x*x) 0.5 m ((x*x) - (16*x) + 64) 0.5 m (2*x -16) 0.5m
5 a	***totalSale = 0.0*** ***bonus = 0.0*** ***for i in range(1, 6):*** ***totalSale = float(input("Enter total sales: "))*** ***bonus = 0.2 * totalSale*** ***print(f"Bonus is RM: {bonus}")***	**3m** (initialize counter - 1m, condition-1m, increment-1m) **1m**- input statement totalSale **1m**- perform expression bonus **1m**- output statement bonus

5b	*code = 0 qty = 0*
	totalPrice = 0.0
	code = int(input("Enter destination code: "))
	qty = int(input("Enter tickets quantity: "))
	if code == 1000:
	    *totalPrice = qty * 60.00 * 1.05*
	elif code == 110:
	    *totalPrice = qty * 85.00 * 1.05*
	elif code == 111:
	    *totalPrice = qty * 70.00 * 1.05*
	print(f"Total ticket price is RM: {totalPrice}")
		1/2m- input statement **code**
		1/2m- input statement **qty**
		1/2m- if statement
		1/2m- expression
		1/2m-else if statement
		1/2m- expression
		1/2m-else if statement
		1/2m- expression
		1m- output statement

6	*total = 0.0*
	count = 0
	average = 0.0
	
	sale = float(input("Enter sale amount (0 to stop): "))
	
	while sale != 0:
	    total = total + sale
	    count = count + 1
	    sale = float(input("Enter sale amount (0 to stop): "))
	
	average = total / count
	
	print(f"Total sales: RM {total}")
	print(f"Number of transactions: {count}")
	print(f"Average sale: RM {average}")
		J0.5 - Correct total variable initialization
		J0.5 - Correct count variable initialization
		J0.5 - Correct average variable initialization
		J1 - Correct first input (prime input)
		J2 - Correct while loop with sentinel condition
		J1.5 - Correct accumulation for total
		J1 - Correct counter increment
		J1 - Correct input inside loop
		J1 - Correct average formula
		J1 - Correct all three outputs