STATE TRANSITION TESTING
(Finite State Machine)

Arivuselvan.K

State Transition Testing

It evaluates the functionality of a system by analyzing its response to changes in input conditions or modifications to its current state.

FINITE STATE MACHINE (FSM):

An FSM is a behavioral model whose outcome depends upon both previous and current inputs.

State Transition Diagram

It has four main components:

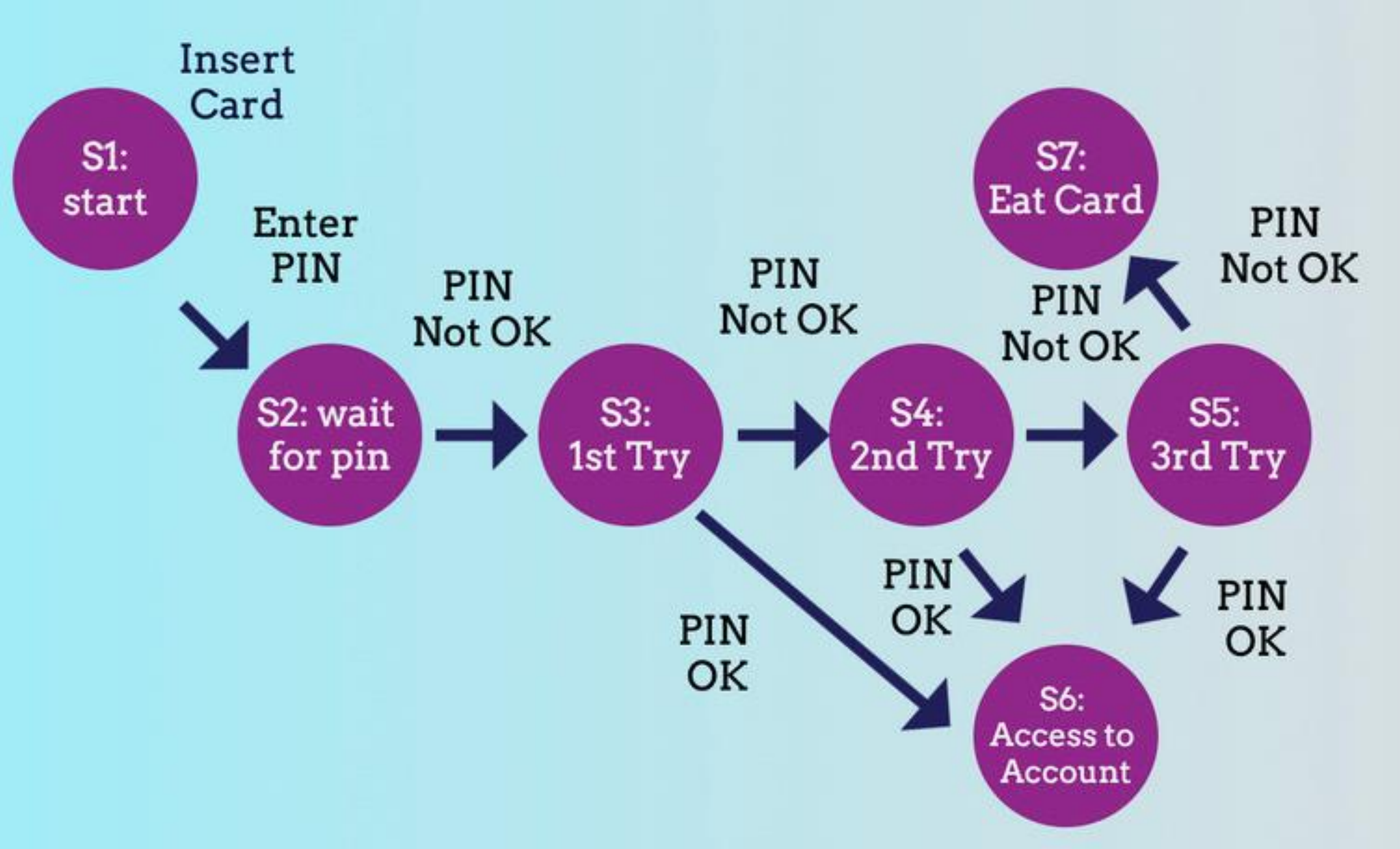
(1) States: State the software current stage.
(example: waiting for the pin, accessing an account)

(2)Transition: A transition is initiated by an event.

(3) Events: The event results in a transition.
(example: Event1: Card inserted, Event 2: enter Pin, Event 3: Pin OK, Event 4: Pin not OK)

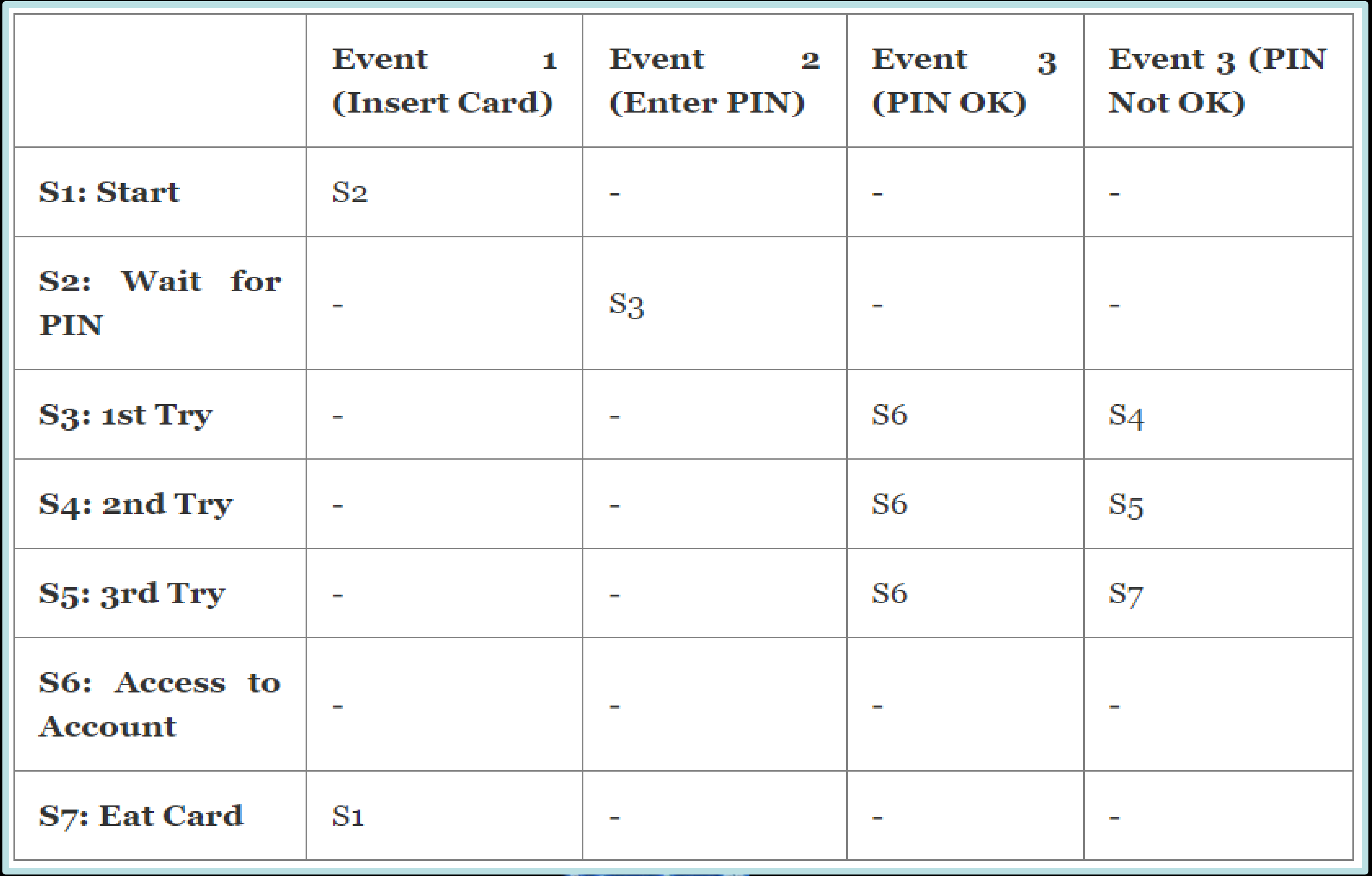
(4) Actions: The state change may result in the software taking action.
(example: Outputting an invalid pin message.)

State Transition Diagram (Example1:ATM Withdrawal process)



State Transition Table

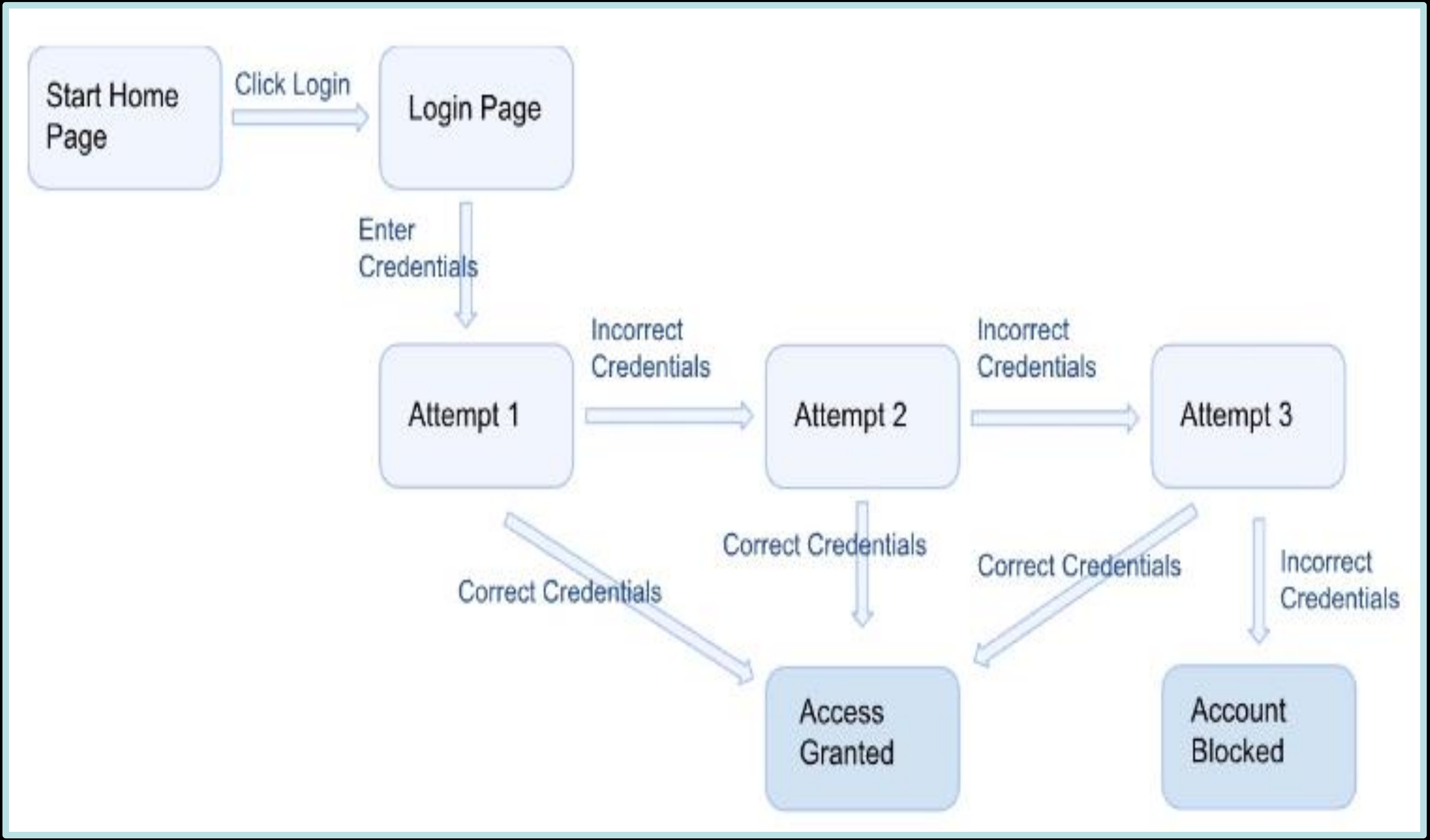
It provides a comprehensive view of all valid and potential invalid transitions between states, (including the events and resulting actions for valid transitions).



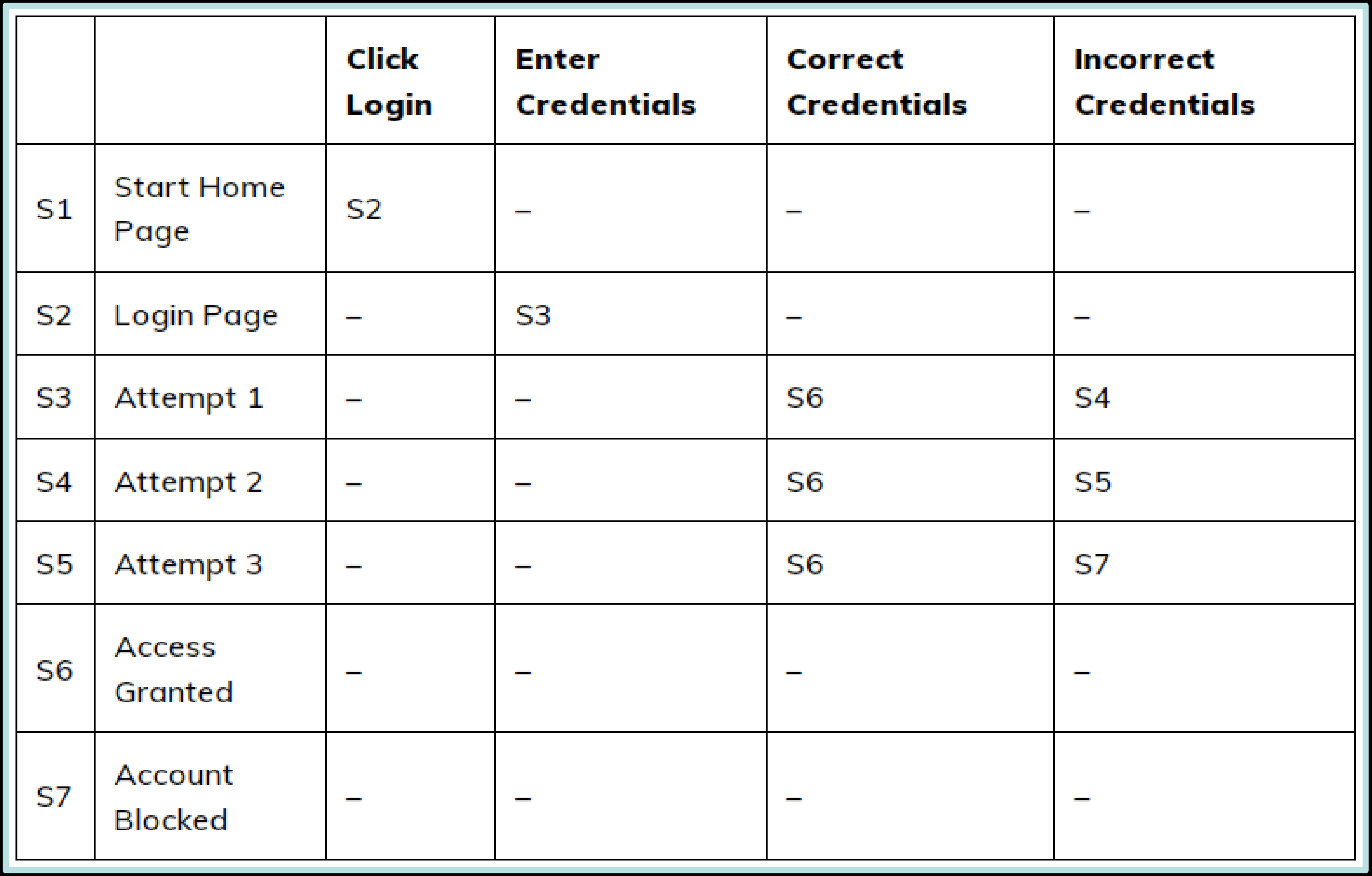
	Event 1 (Insert Card)	Event 2 (Enter PIN)	Event 3 (PIN OK)	Event 3 (PIN Not OK)
S1: Start	S2	-	-	-
S2: Wait for PIN	-	S3	-	-
S3: 1st Try	-	-	S6	S4
S4: 2nd Try	-	-	S6	S5
S5: 3rd Try	-	-	S6	S7
S6: Access to Account	-	-	-	-
S7: Eat Card	S1	-	-	-

Example:2_ Log in to online banking site

State transition diagram:



State Transition Table:



		Click Login	Enter Credentials	Correct Credentials	Incorrect Credentials
S1	Start Home Page	S2	–	–	–
S2	Login Page	–	S3	–	–
S3	Attempt 1	–	–	S6	S4
S4	Attempt 2	–	–	S6	S5
S5	Attempt 3	–	–	S6	S7
S6	Access Granted	–	–	–	–
S7	Account Blocked	–	–	–	–

DECISION TABLE-BASED TESTING

DECISION TABLE-BASED TESTING

Decision table is another useful method to represent the information in a tabular method.

It has to consider complex combinations of input conditions and resulting actions.

Each operand or variable in a logical expression takes on the value, TRUE or FALSE.

DECISION TABLE-BASED TESTING

A **decision table is formed** with the **following components**:

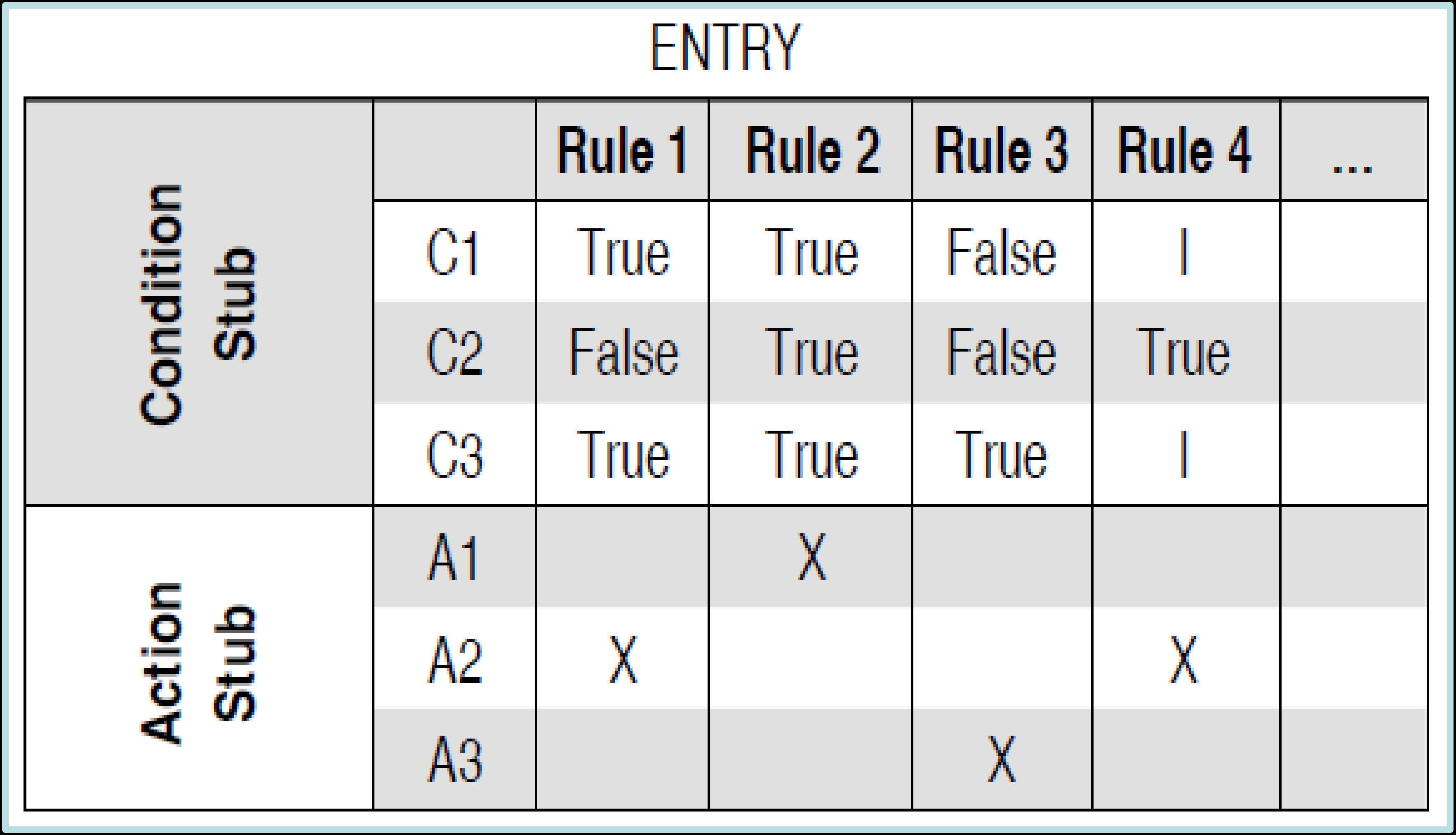
(1) Condition stub: It is a **list of input conditions** for which the **complex combination is made**.

(2) Action stub: It is a **list of resulting actions** which will be performed **if a combination of input condition is satisfied**.

- (‘X’ denotes the **action entry**).
- (‘I’ denotes the **Don’t-Care state or immaterial state**) i.e. **condition entry**, which has **no effect** whether it is **True or False.**

(3) Rule: When we enter **TRUE or FALSE** for all **input conditions for a particular combination**, then it is called a **Rule**.

Decision table structure



ENTRY

Condition Stub		Rule 1	Rule 2	Rule 3	Rule 4	...
	C1	True	True	False	I	
	C2	False	True	False	True	
	C3	True	True	True	I	
Action Stub	A1		X			
	A2	X			X	
	A3			X		

EXAMPLE: 1

The Food delivery company offer the following discounts for their customers based on the given conditions:

New Customer : 15%
Repeat Customer : 10%
Coupon Code : 30%

Design test cases using decision table testing.

Decision Table

		Rules							
		1	2	3	4	5	6	7	8
Condition Stub:	C1: New Customer : 15%	T	T	T	T	F	F	F	F
	C2: Repeat Customer : 10%	T	T	F	F	T	T	F	F
	C3: Coupon Code : 30%	T	F	T	F	T	F	T	F
Action Stub:		Invalid	Invalid	45%	15%	40%	10%	I	I

EXAMPLE: 2

A program calculates the total salary of an employee with the conditions that if the working hours are less than or equal to 48, then give normal salary.

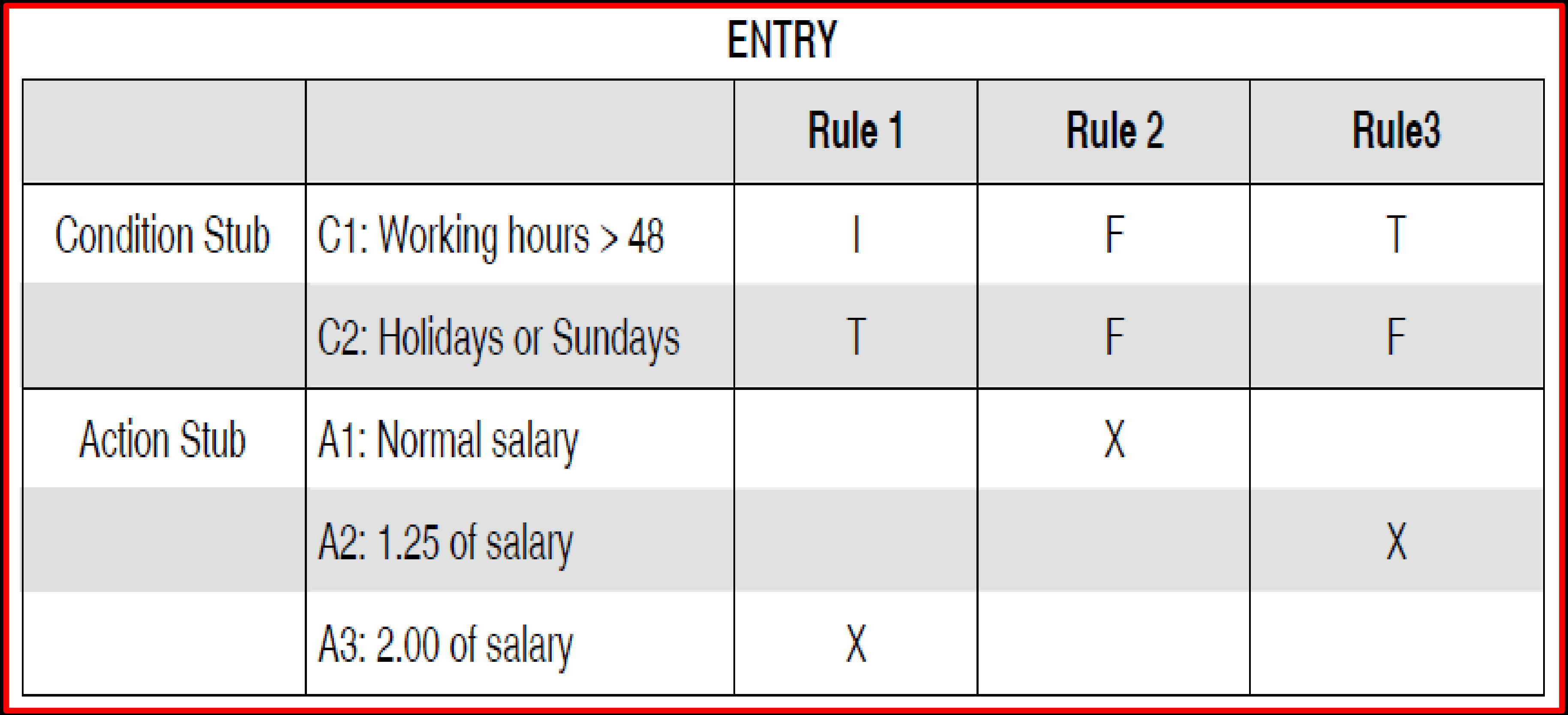
The hours over 48 on normal working days are calculated at the rate of 1.25 of the salary.

However, on holidays or Sundays, the hours are calculated at the rate of 2.00 times of the salary.

Design test cases using decision table testing.

Solution:

The decision table for the program is shown below:

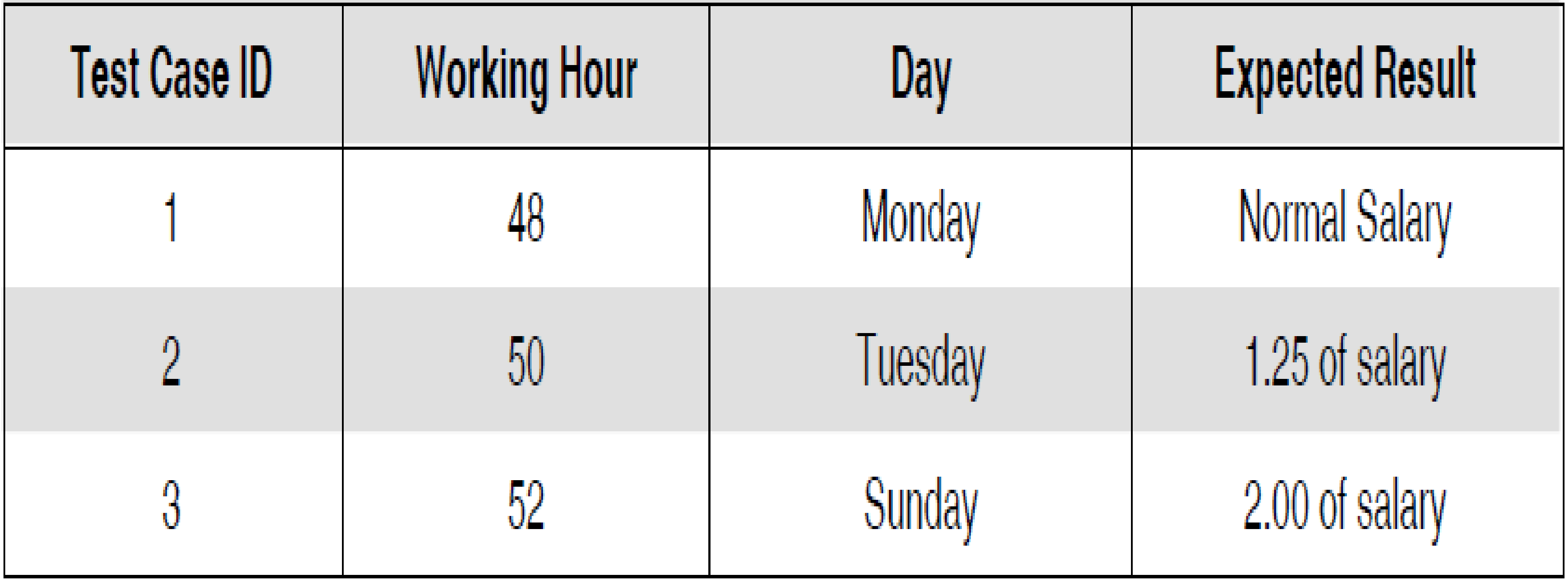


ENTRY

		Rule 1	Rule 2	Rule3
Condition Stub	C1: Working hours > 48	I	F	T
	C2: Holidays or Sundays	T	F	F
Action Stub	A1: Normal salary		X	
	A2: 1.25 of salary			X
	A3: 2.00 of salary	X		

TEST CASE DESIGN USING DECISION TABLE

The test cases derived from the decision table are given below:



Test Case ID	Working Hour	Day	Expected Result
1	48	Monday	Normal Salary
2	50	Tuesday	1.25 of salary
3	52	Sunday	2.00 of salary

EXAMPLE: 3

A university is admitting students in a professional course subject to the following conditions:

(a) Marks in Java ≥ 70

(b) Marks in C++ ≥ 60

(c) Marks in OOAD ≥ 60

(d) Total in all three subjects ≥ 220 OR Total in Java and C++ ≥ 150

If the aggregate mark of an eligible candidate is more than 240, he will be eligible for scholarship course, otherwise he will be eligible for normal course.

The program reads the marks in the three subjects and generates the following outputs:

(i) Not eligible

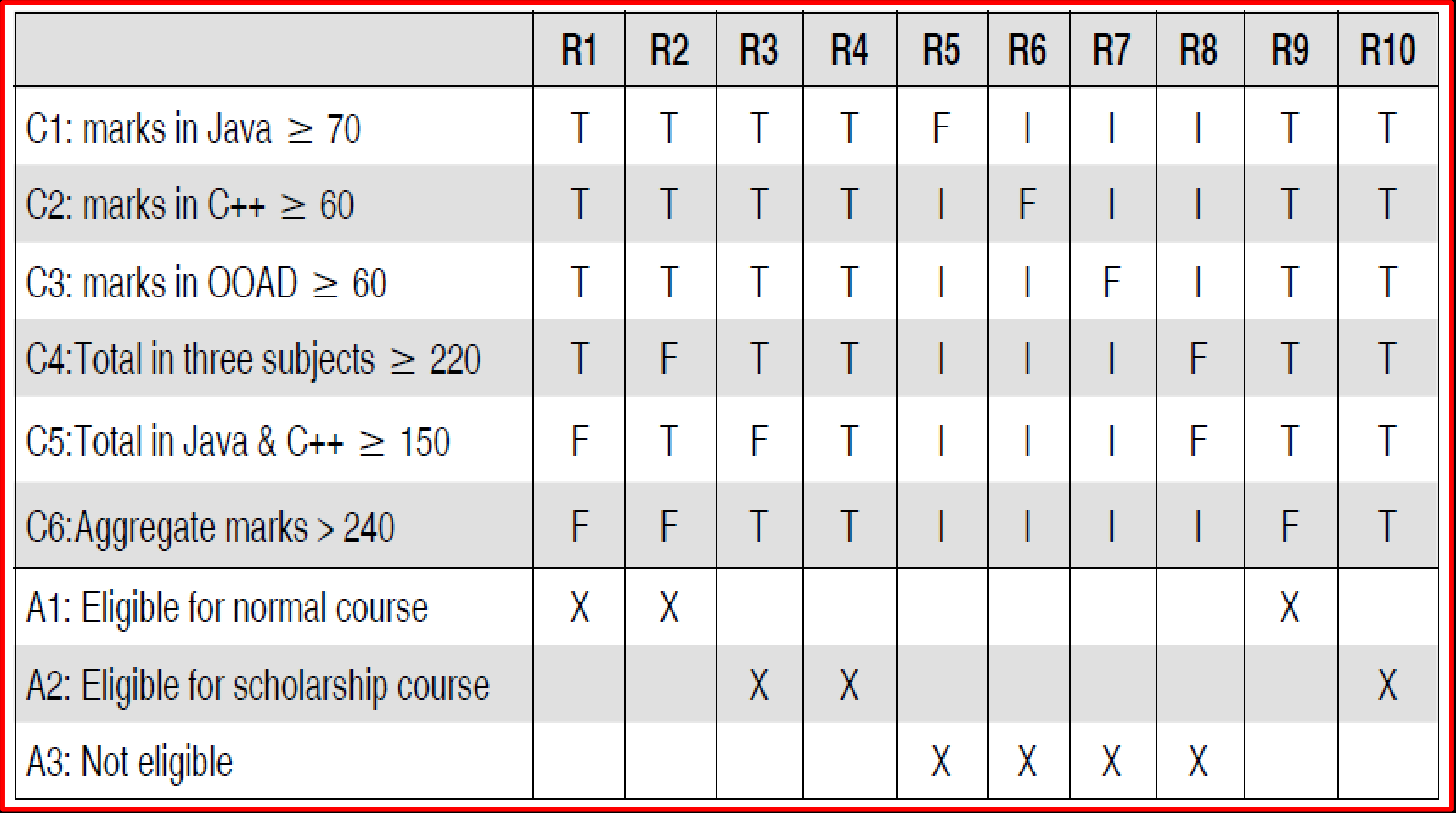
(ii) Eligible for scholarship course

(iii) Eligible for normal course

Design test cases for this program using decision table testing.

Solution:

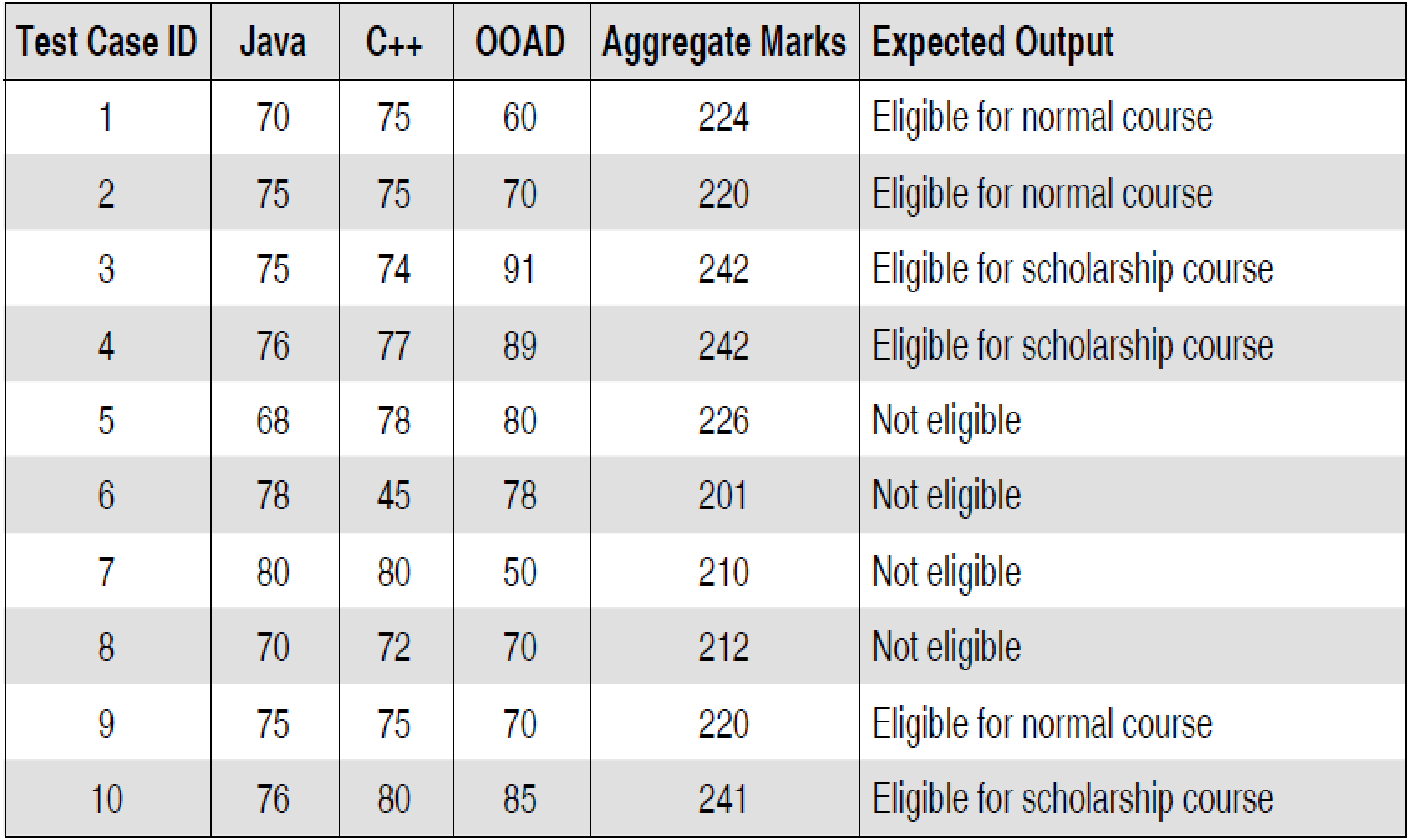
The decision table for the program is shown below:



	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
C1: marks in Java ≥ 70	T	T	T	T	F	I	I	I	T	T
C2: marks in C++ ≥ 60	T	T	T	T	I	F	I	I	T	T
C3: marks in OOAD ≥ 60	T	T	T	T	I	I	F	I	T	T
C4:Total in three subjects ≥ 220	T	F	T	T	I	I	I	F	T	T
C5:Total in Java & C++ ≥ 150	F	T	F	T	I	I	I	F	T	T
C6:Aggregate marks > 240	F	F	T	T	I	I	I	I	F	T
A1: Eligible for normal course	X	X							X	
A2: Eligible for scholarship course			X	X						X
A3: Not eligible					X	X	X	X		

TEST CASE DESIGN USING DECISION TABLE

The **test cases** derived from the **decision table** are given below:



Test Case ID	Java	C++	OOAD	Aggregate Marks	Expected Output
1	70	75	60	224	Eligible for normal course
2	75	75	70	220	Eligible for normal course
3	75	74	91	242	Eligible for scholarship course
4	76	77	89	242	Eligible for scholarship course
5	68	78	80	226	Not eligible
6	78	45	78	201	Not eligible
7	80	80	50	210	Not eligible
8	70	72	70	212	Not eligible
9	75	75	70	220	Eligible for normal course
10	76	80	85	241	Eligible for scholarship course

EXAMPLE: 4

A wholesaler has three commodities to sell and has three types of customers. Discount is given as per the following procedure:

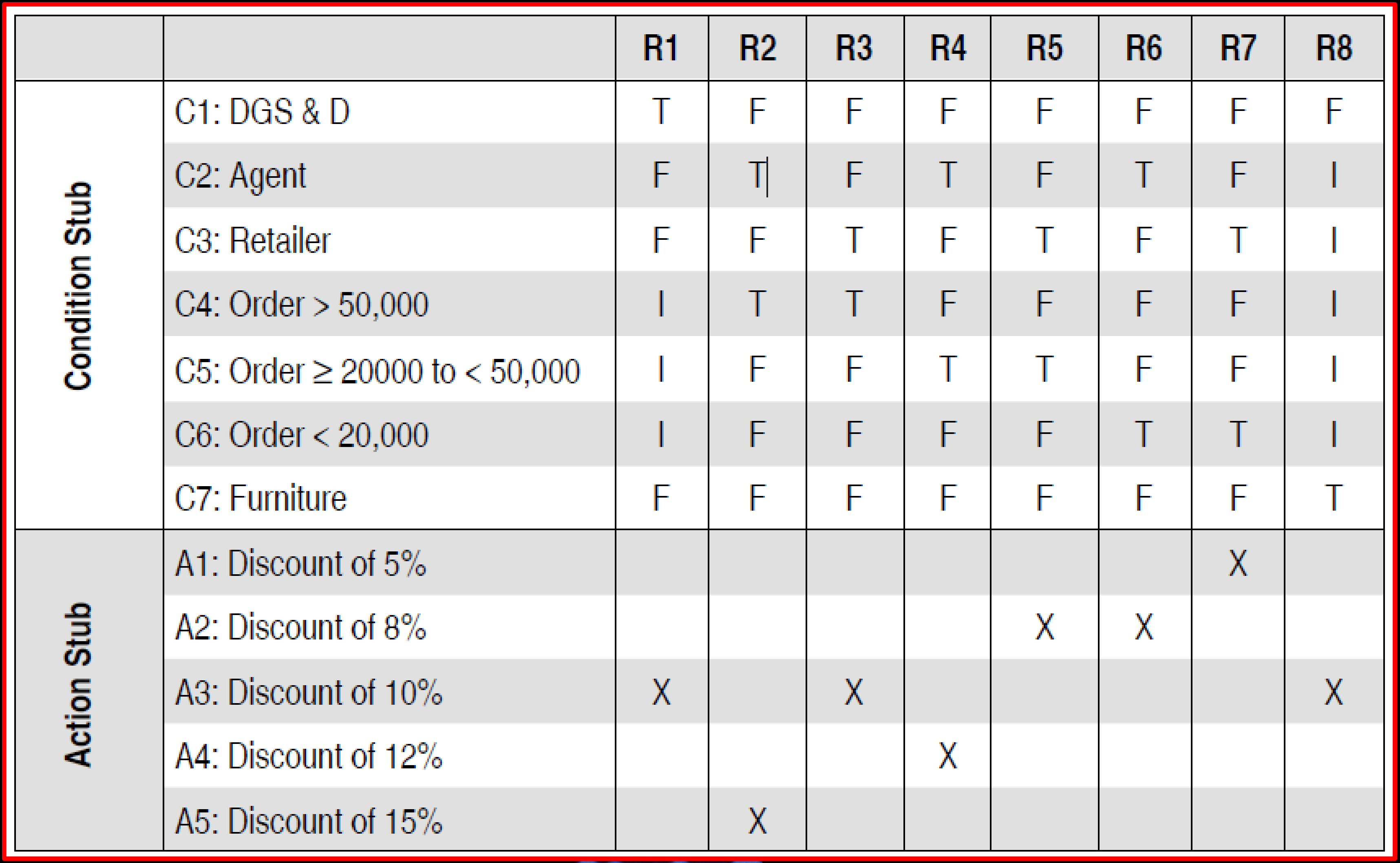
(i) For DGS & D orders, 10% discount is given irrespective of the value of the order.
(ii) For orders of more than Rs 50,000, agents get a discount of 15% and the retailer gets a discount of 10%.
(iii) For orders of Rs 20,000 or more and up to Rs 50,000, agents get 12% and the retailer gets 8% discount.
(iv) For orders of less than Rs 20,000, agents get 8% and the retailer gets 5% discount.

The above rules do not apply to the furniture items wherein a flat rate of 10% discount is admissible to all customers irrespective of the value of the order.

Design test cases for this system using decision table testing.

Solution:

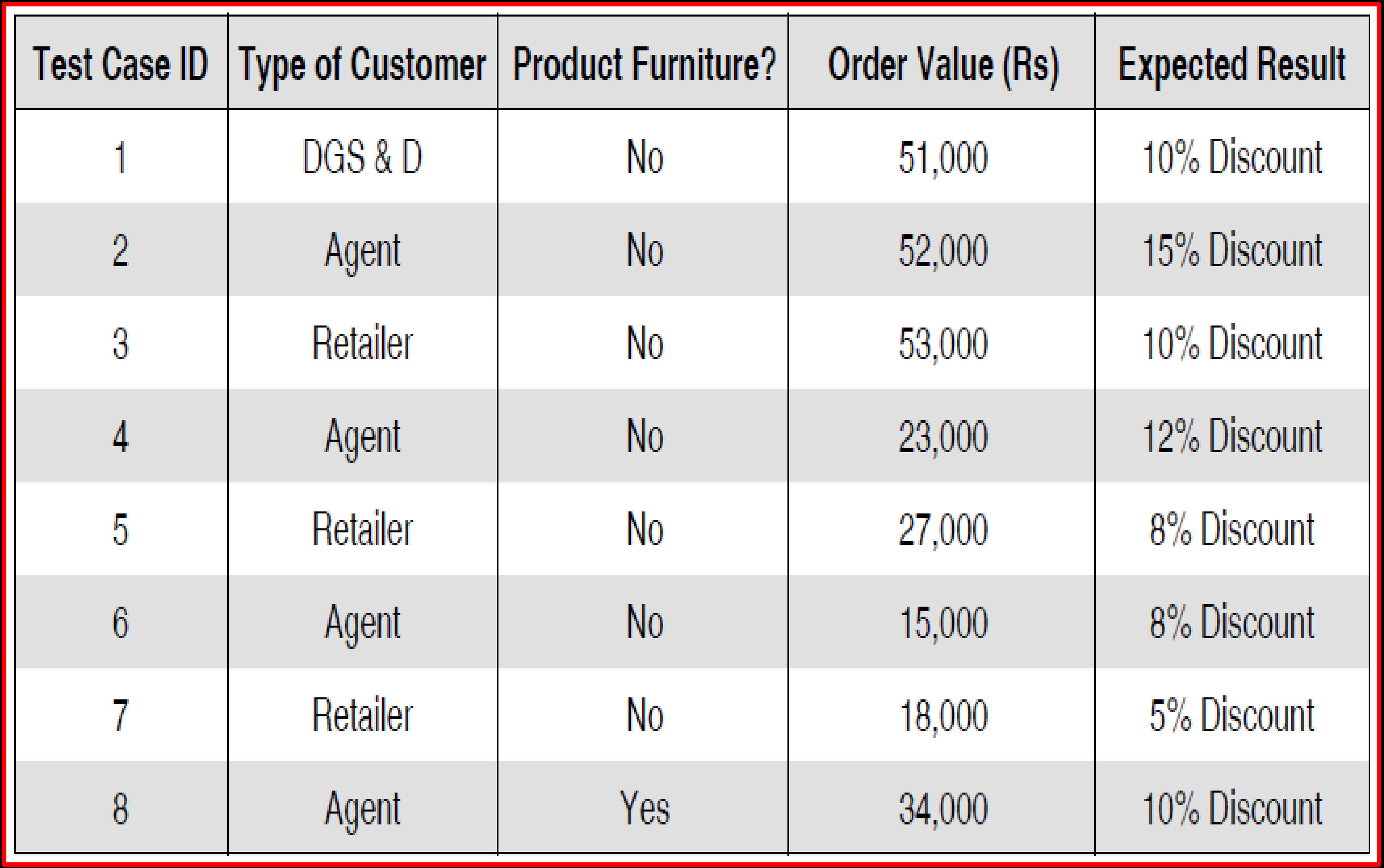
The decision table for the program is shown below:



		R1	R2	R3	R4	R5	R6	R7	R8
Condition Stub	C1: DGS & D	T	F	F	F	F	F	F	F
	C2: Agent	F	T	F	T	F	T	F	I
	C3: Retailer	F	F	T	F	T	F	T	I
	C4: Order > 50,000	I	T	T	F	F	F	F	I
	C5: Order ≥ 20000 to < 50,000	I	F	F	T	T	F	F	I
	C6: Order < 20,000	I	F	F	F	F	T	T	I
	C7: Furniture	F	F	F	F	F	F	F	T
Action Stub	A1: Discount of 5%							X	
	A2: Discount of 8%					X	X		
	A3: Discount of 10%	X		X					X
	A4: Discount of 12%				X				
	A5: Discount of 15%		X						

TEST CASE DESIGN USING DECISION TABLE

The test cases derived from the decision table are given below:



Test Case ID	Type of Customer	Product Furniture?	Order Value (Rs)	Expected Result
1	DGS & D	No	51,000	10% Discount
2	Agent	No	52,000	15% Discount
3	Retailer	No	53,000	10% Discount
4	Agent	No	23,000	12% Discount
5	Retailer	No	27,000	8% Discount
6	Agent	No	15,000	8% Discount
7	Retailer	No	18,000	5% Discount
8	Agent	Yes	34,000	10% Discount