

## **Conditional Statements Exam-21. Multi-Level User Access Control**

### **Description**

A system grants access based on role, account status, and security clearance.

### **Conditions**

- If account inactive → Access Denied: Inactive Account
- Else if role = "Admin" and clearance  $\geq 5$  → Full Access
- Else if role = "Manager" and clearance  $\geq 3$  → Manager Access
- Else if role = "Employee" → Limited Access
- Else → Access Denied

### **Input Format**

role (str)  
is\_active (bool)  
clearance (int)

### **Output Format**

<Access Level>

### **Expected Test Case**

#### **Input**

Admin  
True  
6

#### **Output**

Full Access

## **2. Credit Card Transaction Risk Assessment**

### **Description**

A bank evaluates transaction risk before approval.

## Conditions

- If  $\text{transaction\_amount} > 100000 \rightarrow \text{High Risk}$
- Else if  $\text{international\_transaction}$  and  $\text{amount} > 50000 \rightarrow \text{High Risk}$
- Else if  $\text{failed\_attempts} \geq 3 \rightarrow \text{Transaction Blocked}$
- Else  $\rightarrow \text{Transaction Approved}$

## Input Format

`transaction_amount` (float)  
`international_transaction` (bool)  
`failed_attempts` (int)

## Output Format

`<Transaction Status>`

## Expected Test Case

### Input

60000  
True  
1

### Output

High Risk

## 3. Dynamic Pricing Engine

### Description

An e-commerce platform calculates final price using demand and customer type.

### Conditions

- If  $\text{demand} = \text{"High"}$  and  $\text{customer} = \text{"VIP"}$   $\rightarrow 30\% \text{ markup}$

- If demand = "High" → 20% markup
- If demand = "Medium" → 10% markup
- Else → No markup

### **Input Format**

base\_price (float)  
demand (str)  
customer\_type (str)

### **Output Format**

Final Price: <amount>

### **Expected Test Case**

#### **Input**

1000  
High  
VIP

#### **Output**

Final Price: 1300

## **4. Insurance Premium Classification**

### **Description**

Insurance premium category depends on age, health score, and smoking status.

### **Conditions**

- If smoker and health\_score < 50 → High Premium
- Else if age > 45 and health\_score < 60 → High Premium
- Else if health\_score < 75 → Medium Premium
- Else → Low Premium

### **Input Format**

age (int)  
health\_score (int)  
is\_smoker (bool)

### **Output Format**

<Premium Category>

### **Expected Test Case**

#### **Input**

50  
55  
False

#### **Output**

High Premium

## **5. Smart Inventory Reorder System**

### **Description**

Warehouse system decides reorder priority.

### **Conditions**

- If stock = 0 → Critical Reorder
- Else if stock < minimum and supplier\_available → Immediate Reorder
- Else if stock < minimum → Reorder Pending
- Else → Stock Sufficient

### **Input Format**

stock (int)  
minimum (int)  
supplier\_available (bool)

### **Output Format**

<Reorder Status>

### **Expected Test Case**

#### **Input**

5  
10  
True

#### **Output**

Immediate Reorder

## **6. Employee Performance Action System**

### **Description**

HR decides action based on rating and warnings.

### **Conditions**

- If rating < 2 and warnings  $\geq$  2  $\rightarrow$  Termination
- Else if rating < 3  $\rightarrow$  Performance Improvement Plan
- Else if rating  $\geq$  4 and warnings = 0  $\rightarrow$  Promotion Consideration
- Else  $\rightarrow$  No Action

### **Input Format**

rating (float)  
warnings (int)

### **Output Format**

<Action>

### **Expected Test Case**

#### **Input**

4.5

0

## Output

Promotion Consideration

## 7. API Rate Limiting Controller

### Description

API access is limited based on user tier and request count.

### Conditions

- If requests > 1000 → Blocked
- Else if tier = "Free" and requests > 100 → Rate Limited
- Else if tier = "Pro" and requests > 500 → Rate Limited
- Else → Access Granted

### Input Format

tier (str)  
requests (int)

### Output Format

<API Status>

### Expected Test Case

#### Input

Free  
150

#### Output

Rate Limited

## 8. Loan Interest Rate Decision

## **Description**

Bank assigns interest rate category based on profile.

## **Conditions**

- If  $\text{credit\_score} \geq 750$  and  $\text{salary} \geq 80000 \rightarrow \text{Low Interest}$
- Else if  $\text{credit\_score} \geq 650 \rightarrow \text{Medium Interest}$
- Else  $\rightarrow \text{High Interest}$

## **Input Format**

`credit_score (int)`  
`salary (int)`

## **Output Format**

`<Interest Category>`

## **Expected Test Case**

### **Input**

720  
60000

### **Output**

Medium Interest

## **9. Cloud Resource Auto-Scaling Decision**

## **Description**

Cloud system decides scaling action based on CPU and traffic.

## **Conditions**

- If  $\text{CPU} > 80$  and  $\text{traffic} > 10000 \rightarrow \text{Scale Up Immediately}$
- Else if  $\text{CPU} > 70 \rightarrow \text{Scale Up}$
- Else if  $\text{CPU} < 30 \rightarrow \text{Scale Down}$

- Else → No Scaling

### **Input Format**

cpu\_usage (int)  
traffic (int)

### **Output Format**

<Scaling Action>

### **Expected Test Case**

#### **Input**

85  
12000

#### **Output**

Scale Up Immediately

## **10. Payment Method Selection Engine**

### **Description**

System selects payment method based on amount and wallet balance.

### **Conditions**

- If  $wallet\_balance \geq amount \rightarrow$  Wallet
- Else if  $amount \leq 50000 \rightarrow$  Card
- Else → Net Banking

### **Input Format**

amount (int)  
wallet\_balance (int)

### **Output Format**

<Payment Method>

### **Expected Test Case**

#### **Input**

20000  
5000

#### **Output**

Card

## **11. Customer Support Ticket Priority**

#### **Description**

Support tickets are prioritized dynamically.

#### **Conditions**

- If customer\_type = "Enterprise" and issue\_severity = "Critical" → P1
- Else if issue\_severity = "Critical" → P2
- Else if issue\_severity = "High" → P3
- Else → P4

#### **Input Format**

customer\_type (str)  
issue\_severity (str)

#### **Output Format**

<Ticket Priority>

### **Expected Test Case**

#### **Input**

Enterprise  
Critical

## **Output**

P1

## **12. Student Scholarship Eligibility**

### **Description**

Scholarship eligibility based on marks and income.

### **Conditions**

- If  $\text{marks} \geq 90$  and  $\text{family\_income} < 300000 \rightarrow$  Full Scholarship
- Else if  $\text{marks} \geq 75$  and  $\text{family\_income} < 500000 \rightarrow$  Partial Scholarship
- Else  $\rightarrow$  No Scholarship

### **Input Format**

marks (int)  
family\_income (int)

### **Output Format**

<Scholarship Status>

### **Expected Test Case**

#### **Input**

80  
400000

#### **Output**

Partial Scholarship

## **13. Manufacturing Quality Control**

### **Description**

Product quality grade based on defects and tolerance.

### Conditions

- If defects = 0 → Grade A
- Else if defects ≤ tolerance → Grade B
- Else → Rejected

### Input Format

defects (int)  
tolerance (int)

### Output Format

<Quality Grade>

### Expected Test Case

#### Input

2  
3

#### Output

Grade B

## 14. Delivery Slot Allocation System

### Description

Delivery slot depends on location and urgency.

### Conditions

- If urgent and location = "Local" → Same Day
- Else if urgent → Next Day
- Else → Standard Delivery

### Input Format

urgent (bool)  
location (str)

### **Output Format**

<Delivery Slot>

### **Expected Test Case**

#### **Input**

True  
Local

#### **Output**

Same Day

## **15. System Health Status Analyzer**

### **Description**

Determines system health based on multiple metrics.

### **Conditions**

- If disk < 10 OR memory < 15 → Critical
- Else if disk < 25 OR memory < 30 → Warning
- Else → Healthy

### **Input Format**

disk\_free (int)  
memory\_free (int)

### **Output Format**

<System Status>

### **Expected Test Case**

**Input**

20

40

**Output**

Warning

## 16. Online Examination Proctoring Decision

**Description**

Proctoring system flags candidate behavior.

**Conditions**

- If face\_not\_detected and multiple\_windows → Exam Terminated
- Else if multiple\_windows → Warning Issued
- Else → Exam Normal

**Input Format**

face\_not\_detected (bool)  
multiple\_windows (bool)

**Output Format**

<Exam Status>

**Expected Test Case****Input**

False

True

**Output**

Warning Issued

## **17. Data Backup Policy Selector**

### **Description**

Backup frequency depends on data criticality.

### **Conditions**

- If critical and size > 100 → Hourly Backup
- Else if critical → Daily Backup
- Else → Weekly Backup

### **Input Format**

critical (bool)

data\_size (int)

### **Output Format**

<Backup Policy>

### **Expected Test Case**

#### **Input**

True

120

#### **Output**

Hourly Backup

## **18. Hotel Dynamic Room Pricing**

### **Description**

Room pricing based on season and occupancy.

### **Conditions**

- If season = "Peak" and occupancy > 80 → Surge Pricing

- Else if season = "Peak" → Peak Pricing
- Else → Standard Pricing

### **Input Format**

season (str)  
occupancy (int)

### **Output Format**

<Pricing Type>

### **Expected Test Case**

#### **Input**

Peak  
85

#### **Output**

Surge Pricing

## **19. Fraud Detection Rule Engine**

### **Description**

Detects fraudulent activity.

### **Conditions**

- If location\_mismatch and amount > 50000 → Fraud Detected
- Else if rapid\_transactions ≥ 5 → Suspicious Activity
- Else → Transaction Normal

### **Input Format**

location\_mismatch (bool)  
amount (int)  
rapid\_transactions (int)

## **Output Format**

<Transaction Status>

## **Expected Test Case**

### **Input**

True  
60000  
2

### **Output**

Fraud Detected

## **20. Enterprise Project Risk Evaluation**

### **Description**

Project risk evaluated before approval.

### **Conditions**

- If budget > 10000000 and team\_experience < 3 → High Risk
- Else if deadline < 30 → Medium Risk
- Else → Low Risk

### **Input Format**

budget (int)  
team\_experience (int)  
deadline (int)

### **Output Format**

<Risk Level>

## **Expected Test Case**

### **Input**

15000000

2

45

## **Output**

High Risk