

Customer Loyalty Program – White Box Testing Report

Student Name: HAMZA AHMED SIDDIQUI

Roll No: 22FA-043-SE

Course: SQE (Software Quality Engineering)

Instructor: Sir Talha Ahmed

University: UIT University

Submission Date: 20th November 2025

Full Project: [SQE-REP-GITHUB](#)

1. Program Code (Under Test)

```
loyalty_program.py > calculate_loyalty
1 def calculate_loyalty(current_status, purchase_amount, current_points):
2     """
3     Customer Loyalty Reward Evaluation Algorithm (CLREA)
4     Calculates loyalty points based on purchase value and determines
5     if the customer qualifies for a tier upgrade:
6     Bronze → Silver → Gold.
7     """
8     points_multiplier = 0
9     new_points = current_points
10    new_status = current_status
11    status_change = False
12    # Determine multiplier based on status
13    if current_status == 'Gold':
14        points_multiplier = 2.0
15    elif current_status == 'Silver':
16        points_multiplier = 1.5
17    else:
18        points_multiplier = 1.0
19
20    # Calculate and add points
21    points_earned = int(purchase_amount * points_multiplier)
22    new_points += points_earned
23
24    # Status upgrade conditions
25    if new_points >= 5000:
26        if current_status != 'Gold':
27            new_status = 'Gold'
28            status_change = True
29    elif new_points >= 2000:
30        if current_status not in ('Silver', 'Gold'):
31            new_status = 'Silver'
32            status_change = True
33    elif new_points >= 500:
34        if current_status not in ('Bronze', 'Silver', 'Gold'):
35            new_status = 'Bronze'
36            status_change = True
37
38    if status_change:
39        message = f"Congratulations! You earned {points_earned} points and were upgraded to {new_status} status."
40    else:
41        message = f"You earned {points_earned} points. Your status remains {new_status}."
42    return new_status, new_points, message
43
44 if __name__ == "__main__":
45     print("TC1:", calculate_loyalty("Gold", 100.0, 5100))
46     print("TC2:", calculate_loyalty("Silver", 3500.0, 3000))
47     print("TC3:", calculate_loyalty("Bronze", 2100.0, 0))
48     print("TC4:", calculate_loyalty("Bronze", 100.0, 600))
49     print("TC5:", calculate_loyalty("Newbie", 100.0, 0))
50     print("TC6:", calculate_loyalty("Unranked", 5000.0, 0))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Filter
[Running] python -u "d:\Others\VS CODE\SQE\loyalty_program.py"
TC1: ('Gold', 5300, 'You earned 200 points. Your status remains Gold.')
TC2: ('Gold', 8250, 'Congratulations! You earned 5250 points and were upgraded to Gold status.')
TC3: ('Silver', 2100, 'Congratulations! You earned 2100 points and were upgraded to Silver status.')
TC4: ('Bronze', 700, 'You earned 100 points. Your status remains Bronze.')
TC5: ('Newbie', 100, 'You earned 100 points. Your status remains Newbie.')
TC6: ('Gold', 5000, 'Congratulations! You earned 5000 points and were upgraded to Gold status.')
```

2. Cyclomatic Complexity

Cyclomatic Complexity measures the number of linearly independent paths through the program.

Number of decisions = 5

Cyclomatic Complexity $V(G) = \text{Decisions} + 1 = 5 + 1 = 6$

3. Independent Test Paths

Path 1 → Gold → no upgrade

Path 2 → Silver → upgrade to Gold

Path 3 → Bronze → upgrade to Silver

Path 4 → Bronze → remain Bronze

Path 5 → Unknown status → remain same

Path 6 → Unknown status → upgrade to Gold

4. White-Box Test Cases

TC1: Gold, 5100 pts → no upgrade

TC2: Silver, 3000 pts + 3500 purchase → upgrade to Gold

TC3: Bronze, 2100 purchase → upgrade to Silver

TC4: Bronze, low purchase → remain Bronze

TC5: Newbie, low purchase → remain same

TC6: Unranked, huge purchase → upgrade to Gold

5. Conclusion

The Customer Loyalty Program contains 6 unique execution paths and provides a strong basis for White-Box Testing. All branches were covered using six test cases ensuring full branch coverage and validation of reward logic and status upgrade logic.