

### Question 1:

Assuming you are assigned to conduct the component testing for the method below, please design and create the minimum component test cases needed to archive 100% statement coverage and 100% decision coverage.

```
public class DiscountCalculator {  
    public static double calculateDiscount(double totalAmount) {  
        if (totalAmount >= 100) {  
            return 0.1 * totalAmount; // 10% discount  
        } else {  
            return 0; // No discount  
        }  
    }  
}
```

Note: You must use test case template in the slides to do this exercise.

[illegible]

### Question 2:

Assuming you are assigned to conduct the component testing for the method below, please design and create the minimum component test cases needed to archive 100% statement coverage and 100% decision coverage.

```
1 public class DiscountCalculator {
2     public static double calculateDiscount(double purchaseAmount, boolean isLoyalCustomer) {
3         double discount = 0.0;
4
5         if (purchaseAmount > 100) {
6             discount = 10.0;
7             if (isLoyalCustomer) {
8                 discount += 5.0;
9             }
10        } else if (purchaseAmount > 50) {
11            discount = 5.0;
12        }
13
14        return discount;
15    }
16 }
```

Note: You must use following template to to this question.

[illegible]

### Question 3:

Assuming you are assigned to conduct the component testing for the method below, please design and create the minimum component test cases needed to archive 100% statement coverage and 100% decision coverage.

```

1 public String processOrder(int quantity, boolean isPremiumMember, double totalAmount) {
2     if (quantity <= 0) {
3         return "Invalid quantity";
4     }
5     double discount = 0.0;
6     if (isPremiumMember) {
7         if (totalAmount > 100) {
8             discount = 0.15;
9         } else {
10            discount = 0.10;
11        }
12    } else {
13        if (totalAmount > 200) {
14            discount = 0.05;
15        }
16    }
17    double finalAmount = totalAmount - (totalAmount * discount);
18    if (finalAmount < 50) {
19        return "Order too small to process";
20    }
21    return "Order processed: Final amount = $" + finalAmount;
22 }

```

Note: You must use following template to to this question.

[illegible]

### Question 4:

Assuming you are assigned to conduct the component testing for the method below, please design and create the minimum component test cases needed to archive 100% statement coverage and 100% decision coverage.

```
1 public String analyzeScores(int[] scores) {
2     if (scores == null || scores.length == 0) {
3         return "No scores to analyze";
4     }
5     int sum = 0;
6     int passCount = 0;
7     for (int score : scores) {
8         if (score < 0 || score > 100) {
9             return "Invalid score detected";
10        }
11        sum += score;
12        if (score >= 50) {
13            passCount++;
14        }
15    }
16    double average = sum / (double) scores.length;
17    if (average >= 85 && passCount == scores.length) {
18        return "Excellent performance";
19    } else if (average >= 60) {
20        return "Satisfactory performance";
21    } else {
22        return "Poor performance";
23    }
24 }
```

Note: You must use following template to to this question.

[illegible]