

### 3. Grocery store discount calculator

Input:

```
void main() {
    final Map<String, double> itemPrices = {
        'Rice': 90.00,
        'Dal': 80.00,
        'Flour': 75.00,
        'Oil': 50.00,
        'Milk': 55.00,
        'Sugar': 45.00,
    };

    double totalBeforeDiscount = 0.0;
    itemPrices.forEach((_, price) {
        totalBeforeDiscount += price;
    });

    double discountPercentage;
    if (totalBeforeDiscount < 500.00) {
        discountPercentage = 5.0;
    } else if (totalBeforeDiscount >= 500.00 && totalBeforeDiscount <
1000.00) {
        discountPercentage = 10.0;
    } else {
        discountPercentage = 15.0;
    }

    final double discountAmount = totalBeforeDiscount *
(discountPercentage / 100.0);

    final double finalBill = totalBeforeDiscount - discountAmount;

    print('--- Itemized Bill ---');
    itemPrices.forEach((item, price) {
        print('${item.padRight(12)}: \${price.toStringAsFixed(2)}');
    });
    print('-----');

    print('Total before discount:
\${totalBeforeDiscount.toStringAsFixed(2)}');
```

```

print('Discount (${discountPercentage.toStringAsFixed(0)}%):
\${discountAmount.toStringAsFixed(2)}');
print('Final bill after discount: \${finalBill.toStringAsFixed(2)}');
}

```

Output:

--- Itemized Bill ---

Rice : \$90.00  
 Dal : \$80.00  
 Flour : \$75.00  
 Oil : \$50.00  
 Milk : \$55.00  
 Sugar : \$45.00

-----  
 Total before discount: \$395.00  
 Discount (5%): \$19.75  
 Final bill after discount: \$375.25

#### 4. Ride fare estimator

Input:

```

class Ride {
    final double distance;
    final double ratePerKm;
    final bool peakHour;

    Ride({
        required this.distance,
        required this.ratePerKm,
        this.peakHour = false,
    });

    double calculateFare() {
        final double baseFare = distance * ratePerKm;
        if (peakHour) {
            return baseFare * 1.20;
        }
        return baseFare;
    }
}

void main() {
    final List<Ride> rides = [
        Ride(distance: 10.5, ratePerKm: 2.50, peakHour: false),
        Ride(distance: 5.0, ratePerKm: 2.50, peakHour: true),
    ];
}

```

```

    Ride(distance: 25.3, ratePerKm: 2.50, peakHour: false),
    Ride(distance: 12.0, ratePerKm: 2.50, peakHour: true),
];

double totalEarnings = 0.0;

print(' Ride Fare Estimates ');
for (int i = 0; i < rides.length; i++) {
    final ride = rides[i];
    final double fare = ride.calculateFare();
    totalEarnings += fare;

    print(
        'Ride ${i + 1}: Distance = ${ride.distance.toStringAsFixed(1)} km, '
        'Peak Hour = ${ride.peakHour ? 'Yes' : 'No'}, '
        'Fare = \${fare.toStringAsFixed(2)}',
    );
}

print('Total earnings for all rides:
\${totalEarnings.toStringAsFixed(2)}');
}

```

Output:

```

Ride Fare Estimates
Ride 1: Distance = 10.5 km, Peak Hour = No, Fare = $26.25
Ride 2: Distance = 5.0 km, Peak Hour = Yes, Fare = $15.00
Ride 3: Distance = 25.3 km, Peak Hour = No, Fare = $63.25
Ride 4: Distance = 12.0 km, Peak Hour = Yes, Fare = $36.00
Total earnings for all rides: $140.50

```