

# Budget Management Application

## Scenario:

You have been hired by a small business to create a simple budget management application using Dart.

The application will help the business manage its monthly income and expenses, allowing the user to input various income sources (like sales or investments) and expenses (like rent, salaries, and utilities), then calculate and display the net profit or loss.

## Requirements:

1. Main Method: The application should welcome the user and display a menu.
2. Income and Expenses Management: User can add income, add expense, view net profit/loss, and exit.
3. Data Storage: Use lists to store multiple income and expense entries.
4. Input Validation: Ensure only valid numeric values are entered.
5. Control Flow & Loops: Use if/else statements and loops to manage user choices.
6. String Interpolation & Operations: Display results clearly using string interpolation.
7. Arithmetic Operations: Perform calculations for income, expenses, and profit/loss.
8. Optional Tax Calculation: Apply 15% tax if the business makes a profit.

## Example Output:

```
Welcome to the Budget Management Application!
```

- ```
1. Add income
2. Add expense
3. View summary
4. Exit
```

```
Enter your choice: 1
```

```
Enter income description: Sales
```

```
Enter income amount: 10000
```

```
Income added!
```

```
Enter your choice: 2
```

```
Enter expense description: Rent
```

```
Enter expense amount: 3000
```

Expense added!

Enter your choice: 3

Summary:

Total income: \$10000

Total expenses: \$3000

Net profit: \$7000

Tax to be paid: \$1050

Enter your choice: 4

Exiting the application. Goodbye!

## Dart Code:

```
import 'dart:io';

void main() {
  List<Map<String, double>> incomeList = [];
  List<Map<String, double>> expenseList = [];

  while (true) {
    print("\nWelcome to the Budget Management Application!");
    print("Please choose an option:");
    print("1. Add income");
    print("2. Add expense");
    print("3. View summary");
    print("4. Exit");

    stdout.write("\nEnter your choice: ");
    String? choice = stdin.readLineSync();

    switch (choice) {
      case '1':
        addIncome(incomeList);
        break;
      case '2':
        addExpense(expenseList);
        break;
      case '3':
        viewSummary(incomeList, expenseList);
        break;
      case '4':
        print("Exiting the application. Goodbye!");
        return;
      default:
        print("Invalid choice! Please enter a valid option.");
    }
  }
}
```

```

// Function to add income
void addIncome(List<Map<String, double>> incomeList) {
    stdout.write("Enter income description: ");
    String? description = stdin.readLineSync();

    stdout.write("Enter income amount: ");
    double? amount = double.tryParse(stdin.readLineSync() ?? "");

    if (description != null && amount != null && amount > 0) {
        incomeList.add({description: amount});
        print("Income added!");
    } else {
        print("Invalid input! Please enter a valid amount.");
    }
}

// Function to add expense
void addExpense(List<Map<String, double>> expenseList) {
    stdout.write("Enter expense description: ");
    String? description = stdin.readLineSync();

    stdout.write("Enter expense amount: ");
    double? amount = double.tryParse(stdin.readLineSync() ?? "");

    if (description != null && amount != null && amount > 0) {
        expenseList.add({description: amount});
        print("Expense added!");
    } else {
        print("Invalid input! Please enter a valid amount.");
    }
}

// Function to calculate and display the summary
void viewSummary(List<Map<String, double>> incomeList, List<Map<String, double>> expenseList) {
    double totalIncome = incomeList.fold(0, (sum, item) => sum + item.values.first);
    double totalExpenses = expenseList.fold(0, (sum, item) => sum + item.values.first);
    double netProfit = totalIncome - totalExpenses;
    double tax = (netProfit > 0) ? netProfit * 0.15 : 0;

    print("\nSummary:");
    print("Total income: \${totalIncome.toStringAsFixed(2)}");
    print("Total expenses: \${totalExpenses.toStringAsFixed(2)}");

    if (netProfit >= 0) {
        print("Net profit: \${netProfit.toStringAsFixed(2)}");
        print("Tax to be paid: \${tax.toStringAsFixed(2)}");
    } else {
        print("Net loss: \${(-netProfit).toStringAsFixed(2)}");
    }
}

```

# Smart Home Automation System - Dart

يكد ذيلزنم ةتمتأ ماظنل Dart ةغلل دوك ىلع يوتحي دننسملا اذه  
رماوأل رطس ةهجاو ربع نامأل ماظنو ،ةراجل مظنم ،ءاوضأل في مكحتل احي تي دوكلا  
(CLI).

```
import 'dart:io';

const int minTemp = 15;
const int maxTemp = 30;

void main() {
  bool lightsOn = false;
  int thermostatTemp = 20;
  bool securityArmed = false;

  while (true) {
    print('\nWelcome to the Smart Home System!');
    print('Please choose an action:');
    print('1. Toggle lights');
    print('2. Set thermostat temperature');
    print('3. Arm/disarm security system');
    print('4. View system status');
    print('5. Exit');

    stdout.write('\nEnter your choice: ');
    String? input = stdin.readLineSync();
    int choice = int.tryParse(input ?? '') ?? 0;

    switch (choice) {
      case 1:
        lightsOn = !lightsOn;
        print('\nLights are now ${lightsOn ? 'ON' : 'OFF'}');
        break;

      case 2:
        stdout.write('\nSet thermostat temperature ($minTemp°C - $maxTemp°C): ');
        String? tempInput = stdin.readLineSync();
        int? newTemp = int.tryParse(tempInput ?? '');

        if (newTemp != null && newTemp >= minTemp && newTemp <= maxTemp) {
          thermostatTemp = newTemp;
          print('Thermostat set to $thermostatTemp°C');
        } else {
          print('Invalid temperature! Please enter a value between $minTemp°C and $maxTemp°C');
        }
        break;
    }
  }
}
```

```

case 3:
    securityArmed = !securityArmed;
    print('\nSecurity system is now ${securityArmed ? 'ARMED' : 'DISARMED'}.');
    break;

case 4:
    print('\nSystem Status:');
    print('- Lights: ${lightsOn ? 'ON' : 'OFF'}');
    print('- Thermostat: $thermostatTemp°C ${getTemperatureMessage(thermostatTemp)}');
    print('- Security System: ${securityArmed ? 'ARMED' : 'DISARMED'}');
    break;

case 5:
    print('\nExiting the system. Goodbye!');
    return;

default:
    print('\nInvalid choice! Please enter a number between 1 and 5.');
```

}

}

}

```

String getTemperatureMessage(int temp) {
    if (temp % 3 == 0 && temp % 5 == 0) {
        return '- Perfect Day';
    } else if (temp % 3 == 0) {
        return '- Cool Day';
    } else if (temp % 5 == 0) {
        return '- Warm Day';
    }
    return '';
}

```