

Assignment 1 – Dart Solutions

Includes: Smart Home Automation CLI + Budget Management Application

Part 1: Smart Home Automation System (CLI)

Features: Lights on/off, thermostat (15–30°C) with input validation, security system arm/disarm, status view with FizzBuzz-style messages, and const bounds.

```
import 'dart:io';

void main() {
  // ===== Constants =====
  const int minTemp = 15;
  const int maxTemp = 30;

  // ===== Device states (defaults) =====
  bool lightsOn = false; // lights OFF by default
  int temperature = 20; // thermostat default 20°C
  bool securityArmed = false; // security system DISARMED by default

  while (true) {
    print('\n=== Smart Home System ===');
    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('Choose an option: ');
    String? choice = stdin.readLineSync();

    if (choice == '5') {
      print('Exiting system... Goodbye!');
      break;
    }

    switch (choice) {
      case '1':
        // Toggle lights
        lightsOn = !lightsOn;
        print('Lights are now ${lightsOn ? 'ON' : 'OFF'}.');
        break;

      case '2':
        // Set thermostat temperature with validation
        stdout.write('Enter temperature ($minTemp - $maxTemp): ');
        int? temp = int.tryParse(stdin.readLineSync() ?? '');
        if (temp != null && temp >= minTemp && temp <= maxTemp) {
          temperature = temp;
          print('Temperature set to $temperature°C.');
```

```
        } else {
          print('Invalid temperature! Please try again.');
```

```
        }
        break;

      case '3':
        // Toggle security system
        securityArmed = !securityArmed;
        print('Security system is now ${securityArmed ? 'ARMED' : 'DISARMED'}.');
```

```
        break;
```

```

case '4':
    // Build status strings with interpolation and operations
    String lightsStatus = lightsOn ? 'ON' : 'OFF';
    String securityStatus = securityArmed ? 'armed' : 'disarmed';

    // FizzBuzz-like temperature message
    String tempMessage = '';
    if (temperature % 3 == 0 && temperature % 5 == 0) {
        tempMessage = 'Perfect Day';
    } else if (temperature % 3 == 0) {
        tempMessage = 'Cool Day';
    } else if (temperature % 5 == 0) {
        tempMessage = 'Warm Day';
    }

    print('\n=== System Status ===');
    print('Lights: $lightsStatus');
    print('Temperature: $temperature°C ${tempMessage.isNotEmpty ? "($tempMessage)" : ""}');
    // Convert to upper case when showing status
    print('Security System: ${securityStatus.toUpperCase()}');
    break;

default:
    print('Invalid choice, please try again.');
```

```

    }
}
}

```

Part 2: Budget Management Application (CLI)

Features: Add income/expense (validated, non-negative), lists storage, running summary with net profit/loss, optional 15% tax if profitable (ternary).

```
import 'dart:io';

void main() {
  // Using list of maps: description => amount
  List<Map<String, double>> incomeList = [];
  List<Map<String, double>> expenseList = [];

  while (true) {
    print('\n=== Budget Management System ===');
    print('1. Add Income');
    print('2. Add Expense');
    print('3. View Summary');
    print('4. Exit');
    stdout.write('Choose an option: ');
    String? choice = stdin.readLineSync();

    if (choice == '4') {
      print('Exiting Budget System... Goodbye!');
      break;
    }

    switch (choice) {
      case '1':
        stdout.write('Enter income description: ');
        String desc = stdin.readLineSync()?.trim() ?? 'Income';
        stdout.write('Enter amount: ');
        double? amount = double.tryParse(stdin.readLineSync() ?? '');
        if (amount != null && amount > 0) {
          incomeList.add({desc: amount});
          print('Income added successfully.');
```

```
        } else {
          print('Invalid amount! Please enter a positive number.');
```

```
        }
        break;

      case '2':
        stdout.write('Enter expense description: ');
        String desc = stdin.readLineSync()?.trim() ?? 'Expense';
        stdout.write('Enter amount: ');
        double? amount = double.tryParse(stdin.readLineSync() ?? '');
        if (amount != null && amount > 0) {
          expenseList.add({desc: amount});
          print('Expense added successfully.');
```

```
        } else {
          print('Invalid amount! Please enter a positive number.');
```

```
        }
        break;

      case '3':
        double totalIncome = incomeList.fold(0, (sum, item) => sum + item.values.first);
        double totalExpense = expenseList.fold(0, (sum, item) => sum + item.values.first);
        double net = totalIncome - totalExpense;

        print('\n=== Summary ===');
        print('Total Income: \${totalIncome.toStringAsFixed(2)}');
        print('Total Expense: \${totalExpense.toStringAsFixed(2)}');

        if (net >= 0) {
          print('Your net profit is \${net.toStringAsFixed(2)}');
```

```

        // Optional tax: 15% if profitable, else zero
        double tax = net > 0 ? net * 0.15 : 0;
        if (tax > 0) {
            print('Tax to be paid (15%): \${tax.toStringAsFixed(2)}');
        } else {
            print('No tax due.');
```

```

        }
    } else {
        print('Your net loss is \${net.abs().toStringAsFixed(2)}');
        print('No tax due on losses.');
```

```

    }
    break;
```

```

default:
```

```

    print('Invalid choice! Please select from the menu.');
```

```

}
```

```

}
```

```

}
```