Aim: Understanding and implementing widgets in Flutter

Theory:

Q1. What is Dart Language?

Dart is an object-oriented, open-source programming language developed by Google. It is mainly used for building mobile, web, and desktop applications, especially with the Flutter framework.

Features of Dart:

- Object-oriented programming language
- Supports JIT (Just-In-Time) and AOT (Ahead-Of-Time) compilation
- Simple and readable syntax similar to Java and C++
- High performance with fast execution

Q2. Why Flutter Uses Dart?

- Fast Performance – Dart compiles directly into native machine code
- Hot Reload – Allows instant UI updates without restarting the app
- Single Codebase – Same code works for Android and iOS
- Optimized UI Rendering – Smooth animations and transitions
- Easy to Learn – Clean and beginner-friendly syntax

Q3. What is Widget in Flutter?

A widget is the basic building block of a Flutter application. Everything visible on the screen in Flutter is a widget.

Examples of widgets include:

- Text
- Buttons
- Images
- Layout widgets like Row and Column

Widgets describe how the user interface should look.

Q4. What is Stateless Widget?

A Stateless Widget is a widget whose state does not change during runtime.

Characteristics:

- User interface remains constant
- No internal data change
- Faster performance

Example:

A simple text widget displaying “Hello Flutter”.

Q5. What is Stateful Widget?

A Stateful Widget is a widget whose state can change during runtime.

Characteristics:

- UI changes dynamically
- Uses “setState()” method
- Suitable for interactive applications

Example:

A counter application where the displayed value increases when a button is pressed.

Q6. Steps to develop a Calculator using flutter:

1) Open Android Studio.
2) Create a new Flutter project.
3) Set the project name and location.
4) Open the “main.dart” file.
5) Write the calculator application code using Stateless and Stateful widgets.
6) Use buttons for numbers and operators.
7) Implement basic arithmetic operations.
8) Run the application using an Android emulator.

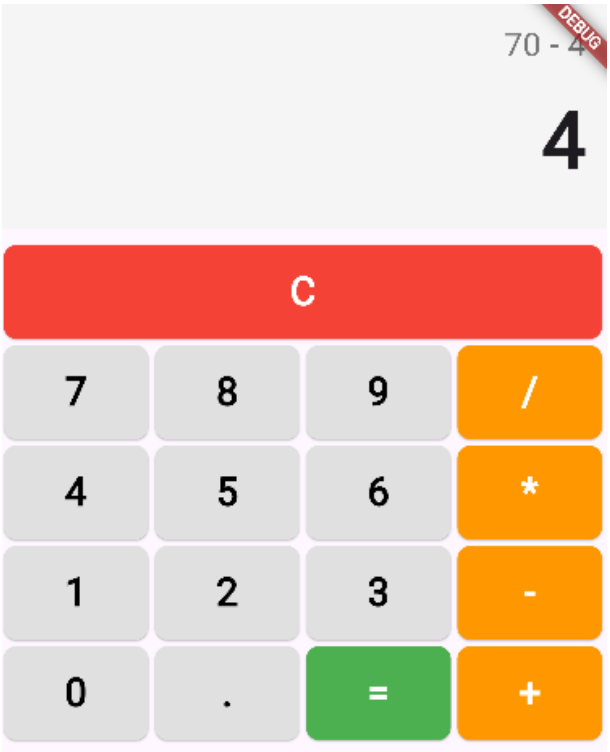
Code (main.dart):

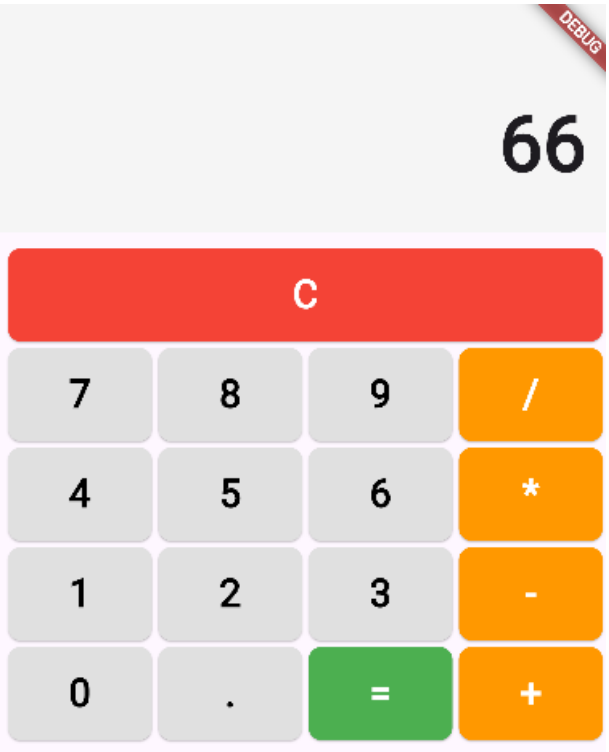
```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Calculator',
      theme: ThemeData(
        colorScheme:
ColorScheme.fromSeed(seedColor:
Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const CalculatorApp(),
    );
  }
}

class CalculatorApp extends
StatefulWidget {
  const CalculatorApp({super.key});

  @override
  State<CalculatorApp> createState() =>
_CalculatorAppState();
}

class _CalculatorAppState extends
State<CalculatorApp> {
  String _display = '0';
  String _expression = '';
  double _firstNumber = 0;
  double _secondNumber = 0;
  String _operation = '';
  bool _shouldClearDisplay = false;

  void _handleButtonPress(String
buttonText) {
```

```
          _operation = '';
          _firstNumber = result;
          _shouldClearDisplay = true;
        }
      } else if (['+', '-', '*',
'/'].contains(buttonText)) {
        // Handle operations
        if (_operation.isEmpty) {
          _firstNumber =
double.tryParse(_display) ?? 0;
          _operation = buttonText;
          _expression = '$_firstNumber
$buttonText ';
          _shouldClearDisplay = true;
        } else {
          // If an operation already
exists, calculate first
          _secondNumber =
double.tryParse(_display) ?? 0;
          double result = _calculate();
          _firstNumber = result;
          _operation = buttonText;
          _expression = '$result
$buttonText ';
          _display = result.toString();
          if (_display.endsWith('.0')) {
            _display =
_display.substring(0, _display.length -
2);
          }
          _shouldClearDisplay = true;
        }
      } else {
        // Handle numbers and decimal
point
        if (_shouldClearDisplay) {
          _display = '';
          _shouldClearDisplay = false;
        }

        if (buttonText == '.') {
          if (!_display.contains('.')) {
            _display += buttonText;
          }
        } else {
          if (_display == '0') {
            _display = buttonText;
          } else {
            _display += buttonText;
          }
        }

        if (_operation.isNotEmpty) {
          _expression = '$_firstNumber
$_operation $_display';
        }
      }
    });
  }

  double _calculate() {
    switch (_operation) {
      case '+':
        return _firstNumber +
_secondNumber;
      case '-':
        return _firstNumber -
_secondNumber;
      case '*':
        return _firstNumber *
_secondNumber;
      case '/':
        return _secondNumber != 0 ?
_firstNumber / _secondNumber : 0;
      default:
        return 0;
    }
  }
          // Calculator buttons
          Expanded(
            child: Padding(
              padding: const
EdgeInsets.all(8.0),
              child: Column(
                children: [
                  // First row: C
                  Row(
                    children: [
                      _buildButton('C',
Colors.red, Colors.white),
                    ],
                  ),

                  // Second row: 7, 8, 9,
/
                  Row(
                    children: [
                      _buildButton('7',
Colors.grey[300]!, Colors.black),
                      _buildButton('8',
Colors.grey[300]!, Colors.black),
                      _buildButton('9',
Colors.grey[300]!, Colors.black),
                      _buildButton('/',
Colors.orange, Colors.white),
                    ],
                  ),

                  // Third row: 4, 5, 6,
*
                  Row(
                    children: [
                      _buildButton('4',
Colors.grey[300]!, Colors.black),
```

```
                    _buildButton('5',
Colors.grey[300]!, Colors.black),
                    _buildButton('6',
Colors.grey[300]!, Colors.black),
                    _buildButton('*',
Colors.orange, Colors.white),
                  ],
                ),

                // Fourth row: 1, 2, 3,
-
                Row(
                  children: [
                    _buildButton('1',
Colors.grey[300]!, Colors.black),
                    _buildButton('2',
Colors.grey[300]!, Colors.black),
                    _buildButton('3',
Colors.grey[300]!, Colors.black),
                    _buildButton('-',
Colors.orange, Colors.white),
                  ],
                ),

                // Fifth row: 0, ., =,
+
                Row(
                  children: [
                    _buildButton('0',
Colors.grey[300]!, Colors.black),
                    _buildButton('.',
Colors.grey[300]!, Colors.black),
                    _buildButton('=',
Colors.green, Colors.white),
                    _buildButton('+',
Colors.orange, Colors.white),
                  ],
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

Ouput:





Conclusion:

Flutter widgets provide a powerful and flexible way to build user interfaces. By using Stateless and Stateful widgets together, interactive applications like a calculator can be developed efficiently.