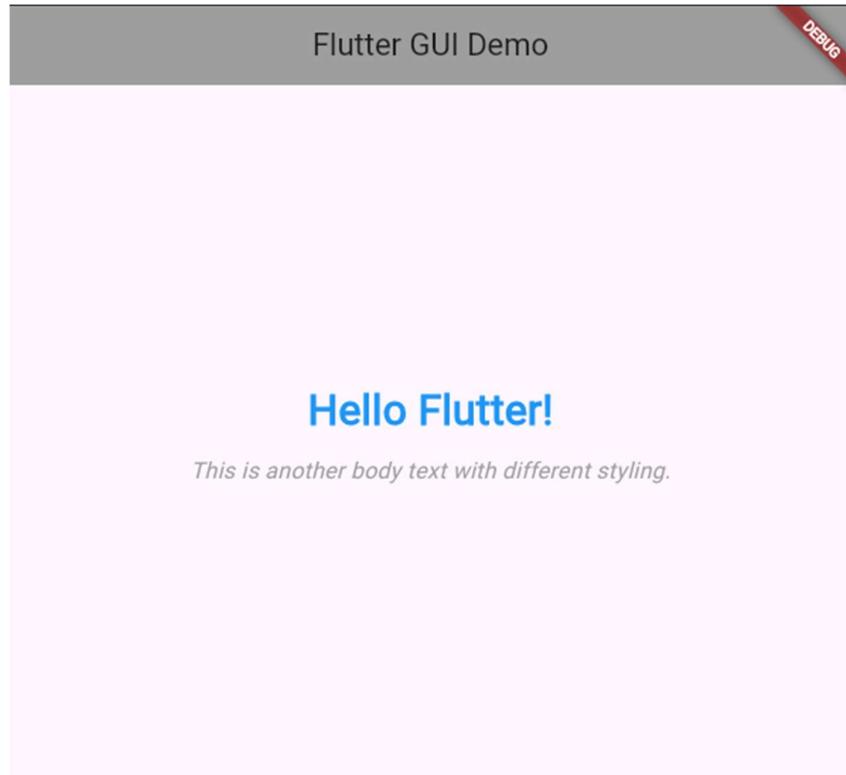


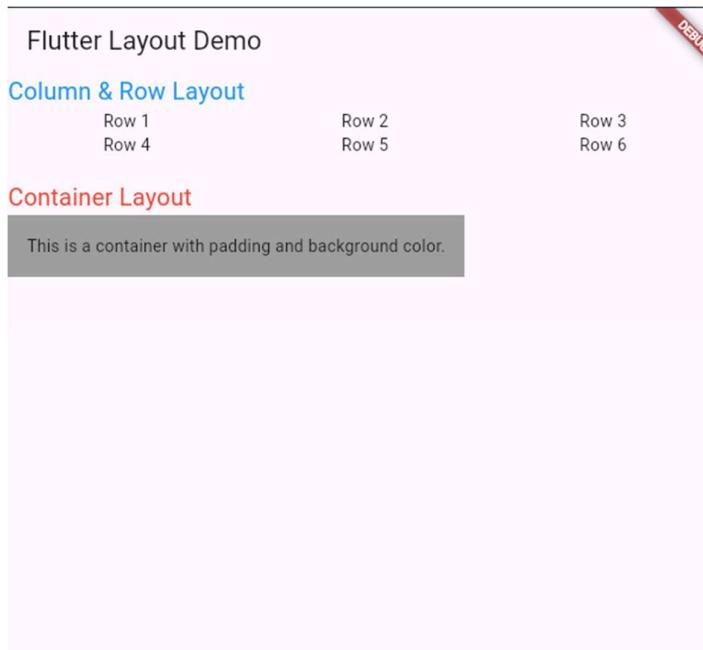
## 1. To develop an application using GUI components, Font and Colours

```
import 'package:flutter/material.dart';
void main() => runApp(
  MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text('Flutter GUI Demo'),
        centerTitle: true,
        backgroundColor: Colors.grey,
      ),
      body: Center(
        child: Column(mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text("Hello Flutter!",style: TextStyle(
            fontSize: 30,
            fontWeight: FontWeight.bold,
            color: Colors.blue),),
          SizedBox(height: 10),
          Text('This is another body text with different styling.',
            style: TextStyle(
              fontSize: 16,
              fontStyle: FontStyle.italic,
              color: Colors.grey),),
        ],),),),
  );
);
```



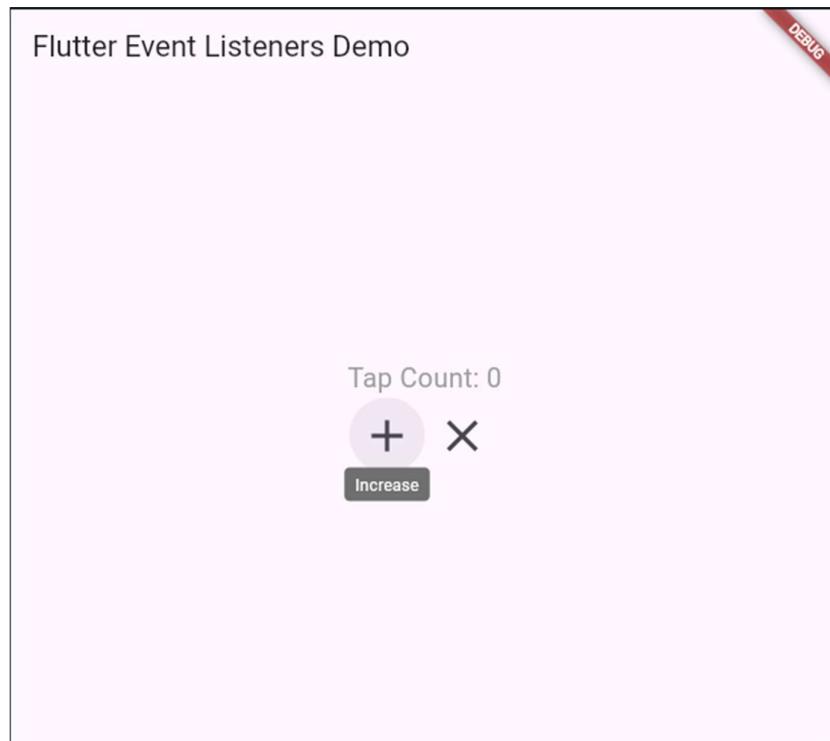
## 2. To develop an application using Layout Managers.

```
import 'package:flutter/material.dart';
void main() => runApp(
  MaterialApp(
    home: Scaffold(
      appBar: AppBar(title: const Text('Flutter Layout Demo')),
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text('Column & Row Layout', style: TextStyle(
            fontSize: 20, color: Colors.blue),),
          Column(crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Row(mainAxisAlignment: MainAxisAlignment.spaceAround,
                children: [Text('Row 1'), Text('Row 2'), Text('Row 3')],
              ),
              Row(mainAxisAlignment: MainAxisAlignment.spaceAround,
                children: [Text('Row 4'), Text('Row 5'), Text('Row 6')],
              ),]),
          SizedBox(height: 20),
          Text('Container Layout',
            style: TextStyle(fontSize: 20, color: Colors.red),),
          Container(padding: EdgeInsets.all(16), color: Colors.grey,
            child: Text('This is a container with padding and background color.')),
        ],
      )));
});
```



### 3. To develop an application for implementing event listeners.

```
import 'package:flutter/material.dart';
void main() => runApp(
  MaterialApp(
    home: Scaffold(
      appBar: AppBar(title: const Text('Flutter Layout Demo'))
      body: Center(
        child: StatefulBuilder(
          builder: (c, set) =>
            Column(mainAxisSize: MainAxisSize.min,
                  children: [Text("Tap Count: $_i",
                  style: TextStyle(
                    fontSize: 20, color: Colors.grey)),),
            Row(mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    IconButton(tooltip: "Increase",
                      icon: Icon(Icons.add,size: 40),onPressed: () => set(() => _i++),
                    ),
                    IconButton(tooltip: "Reset",
                      icon: Icon(Icons.close,size: 40),onPressed: () => set(() => _i=0),
                    ),
                  ],
                )));
  ),
);
int _i = 0;
```



#### 4. To develop an application for drawing basic graphical primitives on the screen

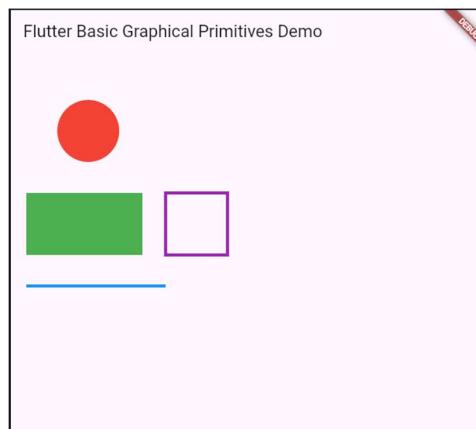
```
import 'package:flutter/material.dart';
void main() => runApp(
  MaterialApp(
    home: Scaffold(
      appBar: AppBar(title: const Text('Flutter Basic Graphical Primitives Demo')),
      body: CustomPaint(
        painter: MyDraw(),
        child: Container(),
      ),),
);
class MyDraw extends CustomPainter {
  @override
  void paint(Canvas c, Size s) {
    var p = Paint()
      ..color = Colors.blue
      ..strokeWidth = 4
      ..style = PaintingStyle.fill;

    var p1 = Paint()
      ..strokeWidth = 4
      ..style = PaintingStyle.stroke;

    c.drawLine(Offset(20, 300), Offset(200, 300),p);
    p.color = Colors.red;
    c.drawCircle(Offset(100, 100),40,p);

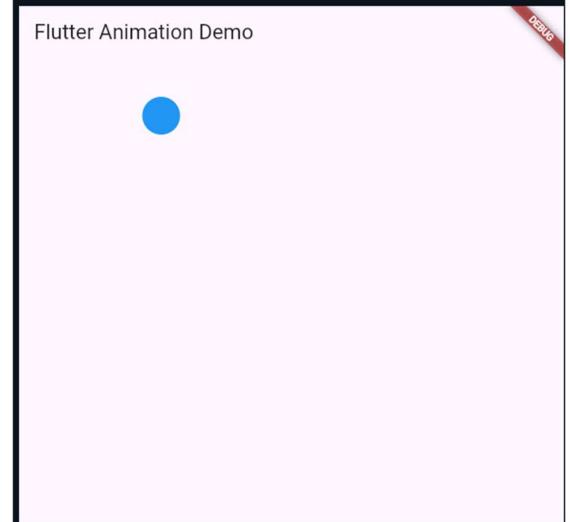
    p.color = Colors.green;
    c.drawRect(Rect.fromLTWH(20, 180, 150, 80),p);

    p1.color = Colors.purple;
    c.drawRect(Rect.fromLTWH(200, 180, 80, 80),p1);
  }
  @override
  bool shouldRepaint(covariant CustomPainter old) => false;
}
```



## 5. To develop an application using animation

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: AutoMoveCircle(),
    );
  }
}
class AutoMoveCircle extends StatefulWidget {
  const AutoMoveCircle({super.key});
  @override
  State<AutoMoveCircle> createState() => _AutoMoveCircleState();
}
class _AutoMoveCircleState extends State<AutoMoveCircle> {
  bool moveRight = false;
  @override
  void initState() {
    super.initState();
    _startAnimation();
  }
  void _startAnimation() async {
    while (true) {
      await Future.delayed(const Duration(seconds: 1));
      setState(() {
        moveRight = !moveRight;
      });
    }
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Flutter Animation Demo")),
      body: Center(
        child: Stack(
          children: [
            AnimatedPositioned(
              duration: const Duration(seconds: 2),
              left: moveRight ? 300 : 0,
              top: 40,
              child: Container(width: 40, height: 40, decoration:
                const BoxDecoration(color: Colors.blue, shape: BoxShape.circle),
            ),
          ],
        ),
      ),
    );
  }
}
```



## **6. To develop an alarm application using flutter.**

```
import 'dart:async';
import 'package:flutter/material.dart';
void main() => runApp(const MyApp());
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      debugShowCheckedModeBanner: false,
      home: AlarmApp(),
    );
  }
}
class AlarmApp extends StatefulWidget {
  const AlarmApp({super.key});
  @override
  State<AlarmApp> createState() => _AlarmAppState();
}
class _AlarmAppState extends State<AlarmApp> {
  TimeOfDay? alarmTime;
  String status = "No Alarm Set";
  Timer? timer;
  void setAlarm() async {
    alarmTime = await showTimePicker(
      context: context,
      initialTime: TimeOfDay.now(),
    );
    if (alarmTime != null) {
      status = "Alarm set for ${alarmTime!.format(context)}";
      checkAlarm();
      setState(() {});
    }
  }
  void checkAlarm() {
    timer?.cancel();
    timer = Timer.periodic(const Duration(seconds: 1), (t) {
      final now = TimeOfDay.now();
      if (now.hour == alarmTime!.hour &&
          now.minute == alarmTime!.minute) {
        t.cancel();
        showDialog(
          context: context,
          builder: (_) => AlertDialog(
            title: const Text("⌚ Alarm"),
            content: const Text("Wake up!"),
            actions: [TextButton(

```

```

        onPressed: () => Navigator.pop(context),
        child: const Text("OK"),],),);
    }
});
}
@override
void dispose() {
    timer?.cancel();
    super.dispose();
}
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: const Text("Simple Alarm Clock")),
        body: Center(
            child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                    Text(status, style: const TextStyle(fontSize: 18)),
                    const SizedBox(height: 20),
                    ElevatedButton(onPressed: setAlarm,
                        child: const Text("Set Alarm"),],),));
    }
}

```

Flutter Alarm Clock Demo

Alarm set for 10:45 AM

[Set Alarm](#)

Flutter Alarm Clock Demo

No Alarm Set

[Set Alarm](#)

Runtimme varinne

Flutter Alarm Clock Demo

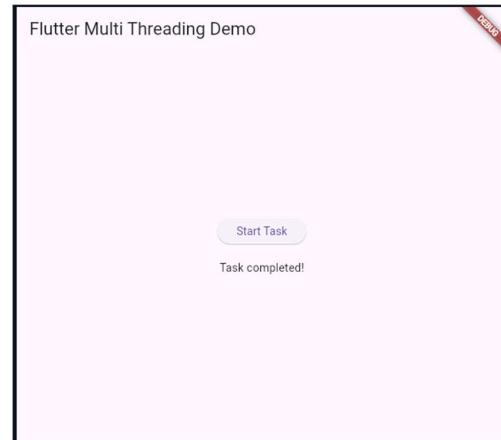
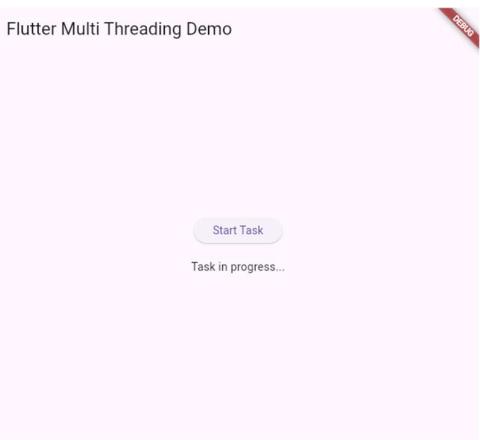
Alarm

Wake up!

OK

## 7. To develop an application using multi-threading

```
import 'package:flutter/material.dart';
void main() => runApp(const MyApp());
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: MultiThread(),);
  }
}
class MultiThread extends StatefulWidget {
  const MultiThread({super.key});
  @override
  State<MultiThread> createState() => _MultiThreadState();
}
class _MultiThreadState extends State<MultiThread> {
  String text = "Press the button";
  void startTask() async {
    setState(() => text = "Task in progress... ");
    await Future.delayed(const Duration(seconds: 5));
    setState(() => text = "Task completed!");
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Flutter Multi Threading Demo")),
      body: Center(
        child: Column(mainAxisAlignment: MainAxisAlignment.center,
        children: [
          ElevatedButton(onPressed: startTask, child: const Text("Start Task"),),
          const SizedBox(height: 20),
          Text(text),],),);
  }
}
```



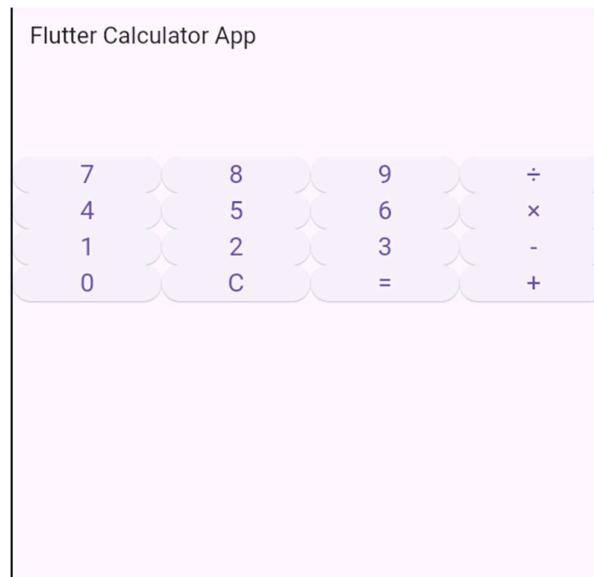
## 8. To develop an application for Calculator

```
import 'package:flutter/material.dart';
void main() => runApp(const MyApp());
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Calculator(),
    );
  }
}
class Calculator extends StatefulWidget {
  const Calculator({super.key});
  @override
  State<Calculator> createState() => _CalculatorState();
}
class _CalculatorState extends State<Calculator> {
  String display = "";
  double num1 = 0;
  String op = "";
  void press(String value) {
    setState(() {
      if (value == "C") {
        display = "";
        num1 = 0;
        op = "";
      } else if (value == "+" || value == "-" || value == "x" || value == "÷") {
        num1 = double.parse(display);
        op = value;
        display = "";
      } else if (value == "=") {
        double num2 = double.parse(display);
        if (op == "+") display = (num1 + num2).toString();
        if (op == "-") display = (num1 - num2).toString();
        if (op == "x") display = (num1 * num2).toString();
        if (op == "÷") display = (num1 / num2).toString();
      } else {
        display += value;
      }
    });
  }
  Widget btn(String text) {
    return Expanded(
      child: ElevatedButton(
        onPressed: () => press(text),
```

```

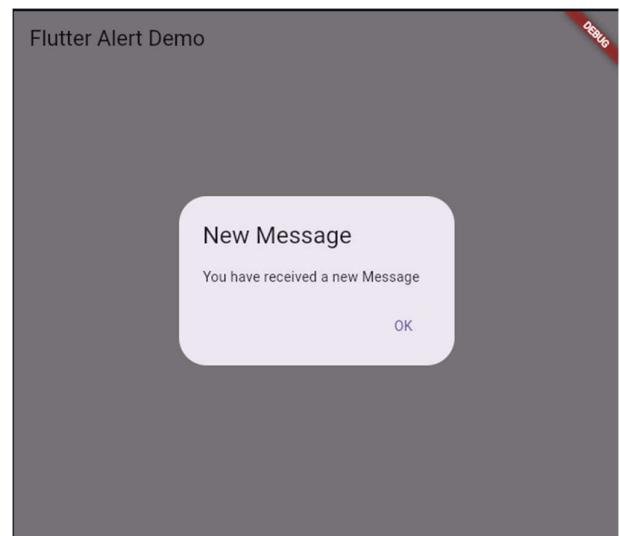
        child: Text(text, style: const TextStyle(fontSize: 24)),
    ),
);
}
@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: const Text("Flutter Calculator App")),
        body: Column(
            children: [
                Container(
                    alignment: Alignment.centerRight,
                    padding: const EdgeInsets.all(20),
                    child: Text(display, style: const TextStyle(fontSize: 32)),
                ),
                Row(children: [btn("7"), btn("8"), btn("9"), btn("÷")]),
                Row(children: [btn("4"), btn("5"), btn("6"), btn("×")]),
                Row(children: [btn("1"), btn("2"), btn("3"), btn("-")]),
                Row(children: [btn("0"), btn("C"), btn("="), btn("+")]),
            ],
        ),
    );
}
}

```



**9. To develop an application for creating an alert upon receiving a message.**

```
import 'package:flutter/material.dart';
void main() => runApp(const MyApp());
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: AlertDemo(),
    );
  }
}
class AlertDemo extends StatelessWidget {
  const AlertDemo({super.key});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Flutter Alert Demo")),
      body: Center(
        child: ElevatedButton(
          child: const Text("Show Alert"),
          onPressed: () {
            showDialog(context: context,
              builder: (_) => AlertDialog(title: const Text("New Message"),
                content: const Text("You have received a new Message"),
                actions: [TextButton(onPressed: () => Navigator.pop(context),
                  child: const Text("OK"),),],));
          },
        );
    );
  }
}
```



## 10. To develop an application for game.

```
import 'dart:math';
import 'package:flutter/material.dart';
void main() => runApp(const MyGame());
class MyGame extends StatelessWidget {
  const MyGame({super.key});
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      debugShowCheckedModeBanner: false,
      home: GameScreen(),
    );
  }
}
class GameScreen extends StatefulWidget {
  const GameScreen({super.key});
  @override
  State<GameScreen> createState() => _GameScreenState();
}
class _GameScreenState extends State<GameScreen> {
  int score = 0;
  double x = 100, y = 200;
  final Random rand = Random();
  void moveBox() {
    setState(() {
      score++;
      x = rand.nextDouble() * 250;
      y = rand.nextDouble() * 500;
    });
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Flutter Tab the Box Game")),
      body: Stack(
        children: [
          Positioned(left: x,top: y,child: GestureDetector(
            onTap: moveBox,child: Container(
              width: 60,height: 60,color: Colors.blue),),),
          Positioned(top: 20,left: 20,child: Text("Score: $score",
            style: const TextStyle(fontSize: 22),),),
        ],
      ),
    );
  }
}
```

Flutter Tab the Box Game

Score: 8

