**Code 8.1:**

import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyModel with ChangeNotifier {  
 int \_count = 0;  
  
 int get count => \_count;  
  
 void increment() {  
 \_count++;  
 notifyListeners();  
 }  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return ChangeNotifierProvider(  
 create: (context) => MyModel(),  
 child: MaterialApp(  
 home: MyHomePage(),  
 ),  
 );  
 }  
}  
  
class MyHomePage extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 final myModel = Provider.of<MyModel>(context);  
  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Provider Package Example'),  
 ),  
 body: Center(  
 child: Text('Count: ${myModel.count}'),  
 ),  
 floatingActionButton: FloatingActionButton(  
 onPressed: () => myModel.increment(),  
 child: Icon(Icons.add),  
 ),  
 );  
 }  
}

**Code 8.2**

import 'package:flutter/material.dart';  
import 'package:flutter\_bloc/flutter\_bloc.dart';  
  
// Events  
abstract class CounterEvent {}  
  
class IncrementEvent extends CounterEvent {}  
  
class DecrementEvent extends CounterEvent {}  
  
// States  
class CounterState {  
 final int count;  
  
 CounterState(this.count);  
}  
  
// BLoC  
class CounterBloc extends Bloc<CounterEvent, CounterState> {  
 CounterBloc() : super(CounterState(0));  
  
 @override  
 Stream<CounterState> mapEventToState(CounterEvent event) async\* {  
 if (event is IncrementEvent) {  
 yield CounterState(state.count + 1);  
 } else if (event is DecrementEvent) {  
 yield CounterState(state.count - 1);  
 }  
 }  
}  
  
// UI  
class CounterApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: BlocProvider(  
 create: (context) => CounterBloc(),  
 child: CounterScreen(),  
 ),  
 );  
 }  
}  
  
class CounterScreen extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 final CounterBloc counterBloc = BlocProvider.of<CounterBloc>(context);  
  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('BLoC Counter App'),  
 ),  
 body: BlocBuilder<CounterBloc, CounterState>(  
 builder: (context, state) {  
 return Center(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text('Count: ${state.count}'),  
 SizedBox(height: 16),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 FloatingActionButton(  
 onPressed: () => counterBloc.add(IncrementEvent()),  
 child: Icon(Icons.add),  
 ),  
 SizedBox(width: 16),  
 FloatingActionButton(  
 onPressed: () => counterBloc.add(DecrementEvent()),  
 child: Icon(Icons.remove),  
 ),  
 ],  
 ),  
 ],  
 ),  
 );  
 },  
 ),  
 );  
 }  
}  
  
void main() {  
 runApp(CounterApp());  
}

**Code 8.3:**

dev\_dependencies:  
flutter\_test:  
sdk: flutter

**Code 8.4:**

// Import necessary packages  
import 'package:flutter\_test/flutter\_test.dart';  
  
// Import the code you want to test  
import 'package:your\_flutter\_project/example.dart'; // Replace with your actual file path  
  
void main() {  
 test('Addition test', () {  
 // Arrange  
 int result = add(2, 3);  
  
 // Assert  
 expect(result, 5);  
 });  
  
 test('Subtraction test', () {  
 // Arrange  
 int result = subtract(5, 2);  
  
 // Assert  
 expect(result, 3);  
 });  
}