

**Assignment No. 1**

**Title:** To Study Basic of Dart Language and Design Basic Flutter App.

**Theory:**

Dart is an open-source, general-purpose programming language developed by Google. It is designed for building high-performance applications, particularly for mobile, web, and desktop platforms. Dart is the primary language used in the Flutter framework, which is a popular UI toolkit for creating natively compiled applications.



**Features of Dart:**

1. Optimized for UI Development

* Dart’s syntax and features are optimized for building graphical interfaces.
* Supports declarative UI programming with the Flutter framework.

2. Strong and Flexible Typing System

* Dart supports both static and dynamic typing.
* The var, final, and const keywords provide flexibility in defining variables.

3. Ahead-of-Time (AOT) and Just-in-Time (JIT) Compilation

* AOT compilation enhances performance by converting code into machine code before execution.
* JIT compilation allows hot reload, making development faster and more interactive.

4. Asynchronous Programming

* Dart provides async and await keywords for handling asynchronous operations effectively.
* The Future and Stream classes support handling delays and event-driven programming.

5. Garbage Collection and Memory Management-

* Dart has automatic memory management, helping developers focus on application logic rather than manual memory allocation.

6. Object-Oriented and Functional Programming Support-

* Dart supports object-oriented programming with classes, mixins, and interfaces.
* It also has functional programming features like higher-order functions and first-class functions.

**Declaring Variables and Data Types:**

void main() {

var name = 'Dart';

String language = 'Flutter';

int year = 2011;

double version = 2.17;

bool isPopular = true;

print('Language: $name, Framework: $language, Year: $year, Version: $version, Popular: $isPopular');

}

**Functions in Dart:**

void greet(String name) {

print('Hello, $name!');

}

void main() {

greet('Flutter');

}

**Using Classes and Objects:**

class Car {

String brand;

int year;

Car(this.brand, this.year);

void display() {

print('Car Brand: $brand, Year: $year');

}

}

void main() {

Car myCar = Car('Tesla', 2022);

myCar.display();

}

**Flutter:**

Flutter is an open-source UI software development toolkit created by Google. It is used to develop applications for mobile, web, desktop, and embedded devices from a single codebase. Flutter applications are written in Dart and compiled to native code using Dart’s AOT compilation for high performance.



**Features of Flutter:**

* Cross-Platform Development - Write once, run on multiple platforms.
* Hot Reload - See changes instantly without restarting the entire application.
* Customizable UI - Flutter’s rich widget library allows for beautiful designs.
* High Performance - Uses Skia for rendering and Dart’s native compilation for efficiency.
* Strong Community and Support - Backed by Google and a large developer community.

**Flutter Widgets:**

Flutter relies on widgets to build its UI. Here are some commonly used widgets:

* Container - Used for styling and layout adjustments.
* Row & Column - Helps arrange elements horizontally or vertically.
* ListView - A scrollable list of elements.
* Stack - Overlapping widgets for advanced UI.
* GestureDetector - Detects taps and gestures.

**State Management in Flutter:**

Flutter offers various state management solutions:

* setState() - The simplest way to manage UI changes.
* Provider - A recommended approach by Google for scalable apps.
* Bloc (Business Logic Component) - Great for larger applications requiring separation of concerns.
* Riverpod - An improvement over Provider with more flexibility.

**Simple Flutter App:**

**Program:**

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatefulWidget {

  @override

  \_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

  TextEditingController \_controller = TextEditingController();

  String \_displayText = "";

  void \_updateText() {

    setState(() {

      \_displayText = \_controller.text;

    });

  }

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Flutter Demo',

      debugShowCheckedModeBanner: false,

      home: Scaffold(

        appBar: AppBar(

          title: Text('MAD Assignment 1'),

        ),

        body: Padding(

          padding: EdgeInsets.all(16.0),

          child: Column(

            mainAxisAlignment: MainAxisAlignment.center,

            children: [

              TextField(

                controller: \_controller,

                decoration: InputDecoration(

                  border: OutlineInputBorder(),

                  labelText: 'Enter text',

                ),

              ),

              SizedBox(height: 10),

              ElevatedButton(

                onPressed: \_updateText,

                child: Text('Submit'),

              ),

              SizedBox(height: 20),

              Text(

                \_displayText,

                style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),

              ),

            ],

          ),

        ),

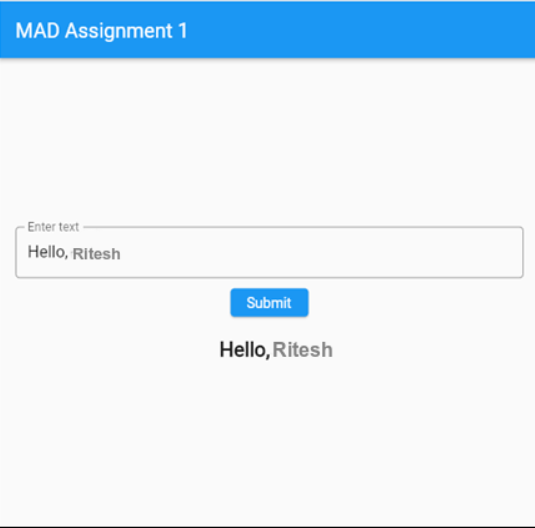
      ),

    );

  }

}

**Output:**

****

**Conclusion:**

Dart is a powerful language optimized for modern app development, and its integration with Flutter makes building cross-platform applications easier and more efficient. By leveraging Dart’s features, developers can create high-performance, visually appealing applications with ease.