WEEK-1

a) Install Flutter and Dart SDK.

Aim: To install and set up Flutter and Dart SDK for developing cross-platform apps on Android, iOS, Web, and Desktop from a single codebase.

Description: Flutter is an open-source UI toolkit by Google that enables developers to build natively compiled applications for mobile, web, and desktop from a single codebase. Dart is the programming language used by Flutter. Installing the Flutter and Dart SDK sets up the development environment needed to create, test, and deploy modern cross-platform applications. This process involves downloading the SDK, configuring system paths, verifying the setup with flutter doctor, and optionally integrating tools like Android Studio or Visual Studio Code for a complete development workflow.

Steps for Installation:

Step 1: Download Flutter SDK
Visit https://flutter.dev
Click the Download Flutter SDK button (ZIP file).
Extract the ZIP file to C:\src\flutter (create the folder if it doesn't exist).

Step 2: Add Flutter to Environment Variables
Press Windows + S and search "Environment Variables".

Click Environment Variables → Under System variables, select Path → Click Edit → New → Add: C:\src\flutter\bin

Step 3: Install Git
Download from https://git-scm.com/download/win
Run the installer with default options.

Step 4: Verify Installation
Open Command Prompt (cmd)
Run:flutter doctor

b) Write a simple Dart program to understand the language fundamentals.

Aim: To write a simple Dart program demonstrating basic language fundamentals such as variables, data types, conditionals, loops, functions, and object-oriented programming.

Description: This Dart program introduces the core concepts of the Dart programming language. It demonstrates the use of variables with different data types (String, int, double, bool), conditional statements (if-else), loops (for-in), functions, and basic object-oriented programming using classes and objects. The program outputs user information, evaluates a pass/fail condition, lists subjects, and shows how to define and use a class with a constructor and method. This foundational program helps beginners understand how Dart syntax and structure work in real-world scenarios.

Source Code:

```
void main() {
  String name = 'Dart Learner';
  int age = 20;
  int a = 10;
  double b = 5.5;
  double score = 85.5;
  bool isPass = true;
  print('Name: $name');
  print('Age: $age');
  print('Score: $score');
  print('Passed: $isPass');

  if (score >= 50) {
    print('Result: Pass');
  } else {
    print('Result: Fail');
  }

print('For loop (1 to 5):');
  for (int i = 1; i <= 5; i++) {
    print('i = $i');
  }

print('While loop (5 to 1):');
  int j = 5;
  while (j >= 1) {
    print('j = $j');
    j--;
  }
```

```
print('Addition: ${a + b}');
  print('Subtraction: ${a - b}');
  print('Multiplication: ${a * b}');
  print('Division: ${a / b}');
  print('Modulus: ${a % b}\n');
```

OUTPUT:

```
Name: Dart Learner
Age: 20
Score: 85.5
Passed: true
Result: Pass
For loop (1 to 5):
i = 1
i = 2
i = 3
i = 4
i = 5
While loop (5 to 1):
j = 5
j = 4
j = 3
j = 2
j = 1
Addition: 15.5
Subtraction: 4.5
Multiplication: 55
Division: 1.8181818181818181
Modulus: 4.5
```