Flutter Assignments

Module 1: Introduction to Mobile Development and Flutter

1. Explain the benefits of using Flutter over other cross-platform frameworks :

* Flutter uses a single language “dart” to code which makes, easy to code over android and IOS , also for web apps .
* Also it performs faster , takes less time to code .

1. Describe the role of Dart in Flutter. What are its advantages for mobile development?

* “Dart” is the single language in flutter which is used for frontend and backend both.
* Dart makes code easy and takes less time
* Dart is allow to use it for cross platforms like android , ios , web app .
* Dart also have many official third party libraries or packages which reduces the code and make seamless UI .

1. Outline the steps to set up a Flutter development environment.

* Install flutter SDK from official website
* Extract that SDK file in c:\program files
* Set the path of bin folder path in Environment variables
* Install IDE (Android studio or VS code)
* In VS code install Flutter and Dart extension to run flutter projects in it .
* In Android Studio install flutter Plugin and dart Plugin

1. Describe the basic Flutter app structure, explaining main.dart, the main function, and the widget tree.

* Basic flutter app structure – flutter app structure is built using main.dart , which contains main function , here’s is the breakdown :
* Main.dart- It is a entry point of flutter application , it conatins main fuction which starts the app
* Main function – It is a function from where the execution begins , it calls runapp() and takes widget as an argument also called root of widget tree.
* Widget tree – Flutter app is made up of many widgets which are in structured manner this should be called widget tree.There are two types of widget in this : Stateless widget (Does not change state) , Statefull widget (It can change state dynamically)

Module2

1. Explain the fundamental data types in Dart (int, double, String, List, Map, etc.) and their uses.

* Int - Represents whole numbers , ex: int = 9 . Used for counting and arithmetic operations.
* Double - Represents decimal numbers, e.g., double price = 19.99; . Used for precise calculations involving fractions.
* String - Represents text, e.g., String name = "Raj";. Used for storing and manipulating characters and words.
* List – Represents an ordered collection of items, e.g., List<int> numbers = [1, 2, 3];. Used for storing multiple values in a sequence.
* Map – Represents key-value pairs, e.g., Map<String, int> scores = {"Alice": 90, "Bob": 85};. Used for fast lookups and data mapping.

1. Describe control structures in Dart with examples of if, else, for, while, and switch.

* The if statement is used to execute a block of code when a condition is true. The else statement provides an alternative execution when the condition is false. For example::

int age = 18;

if (age >= 18) {

print("You are an adult.");

} else {

print("You are a minor.");

}

* A for loop is used to iterate over a sequence of numbers or a collection. It consists of an initialization, condition, and increment/decrement. For example::

for (int i = 0; i < 5; i++) {

print("Iteration $i");

}

* A while loop executes a block of code as long as the specified condition is true. For example::

int count = 0;

while (count < 3) {

print("Count: $count");

count++;

}

* The switch statement is used to execute one block of code from multiple options based on a specific value.

String day = "Monday";

switch (day) {

case "Monday":

print("Start of the week.");

break;

case "Friday":

print("Weekend is near!");

break;

default:

print("A regular day.");

}

1. Explain object-oriented programming concepts in Dart, such as classes, inheritance, polymorphism, and abstraction.

* A **class** is a blueprint for creating objects. It defines properties (variables) and behaviors (methods) that objects of the class can have. Used to represent real-world entities like Users, Products, or Vehicles. For Example::

Class Car {String brand; int speed ;}

* **Inheritance** allows a class to inherit properties and methods from another class, enabling code reusability. The extends keyword is used for inheritance in Dart. Helps avoid code duplication and create hierarchical relationships.
* **Polymorphism** allows a subclass to provide a different implementation of a method that is already defined in its super class. The @**override** annotation is used in Dart. Used when multiple classes share a method but need different
* **Abstraction** hides implementation details and only shows essential features. It is achieved using **abstract classes** or **interfaces** in Dart. Used to define common behaviors that multiple classes should implement