Subject Name: UI/UX Designing using Dart and Flutter Laboratory
Subject Code: BCA606

Sr	Topics Details	date	sign
1	**Introduction to Dart:** 1. Installation of Android Studio and DART SDK. 2. print "Hello, Dart!" 3. declare variables of various types with basic operations. 4. use of var, final, and const in Dart. 5. arithmetic, relational, and logical operators .		
2	**Decision-Making Statements in Dart** 1. accepts three numbers and determines the largest number . 2. checks whether a given year is a leap year or not use if-else 3. student's score and grade 4. accepts a number (1-7) and prints the corresponding week day 5. user's number and checks whether it is even or odd		
3	**Loops in Dart** 1. calculates the sum of first n natural numbers using a for loop. 2. the factorial of a given number using a while loop. 3. Fibonacci series up to n terms using a do-while loop. 4. reverses a given integer using a while loop. 5. displays the multiplication table of a given number (for loop)		
4	**Functions in Dart** 1. function to calculate the area of a circle 2. function to find the maximum of three numbers. 3. function to check if a given number is prime. 4. Function to convert a temperature from Celsius to Fahrenheit. . 5. function to count the number of vowels in a given string.		
5	**Oops in Dart** 1. Create a Dart class Employee with attributes name, age, and salary. Define a method to display employee details. 2. Implement multiple inheritance interfaces. Create an interface Payable and another class Employee that implements Payable. 3. Define method overloading by creating a class Calculator with different methods to add integers, doubles, and strings 4. defines a base class Shape with a method area(). Create two subclasses, Circle and Square, that override the area() method to calculate their respective regions. Demonstrate polymorphism by creating objects of both subclasses and calling the area() method. 5. defines a Student class with private properties for name and age. Include public methods to set and get the values of these properties.		
6	**Packages in Dart** 1. creates a custom package named math_operations that contains a function add(int a, int b) that returns the sum of two integers. Demonstrate how to use this package in another Dart file. 2. defines a geometry package that includes a class Circle with methods to calculate the area and circumference. 3. Write a Dart program that creates a package named math_utilities that includes the following functions: double add(double a, double b) double subtract(double a, double b) double multiply(double a, double b) Demonstrate the use of this package in another Dart file .		

7	**Exceptions in Dart** 1. defines a function divide(int a, int b) that performs division. Implement exception handling to catch a division by zero error 2. Create a function readFile(String filename) that throws an exception if the file does not exist. . 3. Reads an integer from the user and throws an exception if the input is invalid (e.g., a non-integer). 4. defines a custom exception class InvalidAgeException. Create a function checkAge(int age) that throws this exception if the age is less than 0 or greater than 120.		
8	**Dart Collections** 1. Write a Dart program that creates a list of grocery items for a shopping list. Implement the following functionalities: • Add new items to the list. • Remove items from the list. • Display the current items in the shopping list. 2. Write a Dart program that counts the frequency of each word in a given string using a map. The program should output each word along with its corresponding count. **Expected Input:** String: "hello world hello" **Expected Output:** hello: 2 world: 1 3. Write a Dart program that takes a list of integers with possible duplicates and returns a set of unique integers. Demonstrate this by creating a list, converting it to a set, and printing the unique elements. **Expected Input:** • List: [1, 2, 3, 2, 1, 4, 5, 4] **Expected Output:** • Unique elements: {1, 2, 3, 4, 5}		
9	**Flutter Widgets:** 1. Write a Program to Create a simple Flutter application that displays a counter. Implement a button that increments the counter when pressed. Use Text, ElevatedButton, and StatefulWidget to build the application. 2. Create a Flutter application that displays a list of items using ListView. Populate the list with at least five strings (e.g., fruits or vegetables). 3. Create a Flutter application that contains a form with a TextField for user input and a button. When the button is pressed, display the input value on the screen. 4. Write a Program to Create a Flutter application that displays an image from the internet using Image.network(). Provide a placeholder while the image loads. 5. Create a Flutter application that displays an image gallery using GridView. Use a list of image URLs to populate the grid.		
10	**App's Navigation and Database Connectivity** 1. Write a Program to Create a Flutter application with two screens: a home screen and a details screen. The home screen should have a button that, when pressed, navigates to the details screen. Pass a message from the home screen to the details screen and display it. **Expected Output:** Home screen with a button labeled "Go to Details." Details screen displaying the passed message.		

	2. Write a Program to Create a Flutter application that uses named routes for navigation. Set up three screens: home, settings, and about. Implement buttons on the home screen that navigate to the settings and about screens using named routes. **Expected Output:** Home screen with buttons to navigate to Settings and About screens. 3. Create a Flutter application with two screens where the first screen has a list of items. When an item is tapped, it navigates to the second screen that displays the item details. Implement a back navigation that returns the user to the first screen. **Expected Output:** First screen with a list of items. Second screen displaying item details and a back button.		
11	**Basic App Development – HelloWorld App, To-Do App** 1. Write a Program to Create a simple Flutter application that displays "Hello, World!" on the screen. Use basic widgets like Text and Center to format the display. 2. Write a Program to Create a basic To-Do app that allows users to add and display a list of tasks. Implement a TextField for task input and a button to add the task to the list. Display the tasks using a ListView.		
12	**Basic App Development – Simple Calculator App, Unit Converter App** 1. Write a Program to Create a Flutter application that functions as a basic calculator. The app should allow users to perform addition, subtraction, multiplication, and division operations. 2. Write a Program to Create a Flutter application that converts units from one type to another (e.g., length, weight, temperature). Implement conversion for at least three types of units.		
13	**Basic App Development – BMI Calculator App, Habit Tracker App** 1. Create a Flutter app that calculates the Body Mass Index (BMI) based on user input for weight (in kilograms) and height (in meters). Display the calculated BMI along with the corresponding category (Underweight, Normal, Overweight, Obesity). 2. Create a Flutter app that allows users to add daily habits. Each habit should have a title and a checkbox to mark it as complete.		
14	**Basic App Development – Weather App, Login/Signup App** 1. Create a Flutter app that displays the current weather information for a specified city using a public weather API (like OpenWeatherMap). 2. Enhance the Weather App to fetch and display weather information based on the user's current location		
15	**Basic App Development – Recipe App, Quiz App** 1. Write a Program to Create a Flutter app that displays a list of recipes with titles, images, and brief descriptions. Users should be able to tap on a recipe to view more details. **2.** Write a Program to Enhance the Recipe App to include a search feature that allows users to filter recipes by name or ingredient.		