COURSE TITLE	**LANGUAGE FOR CROSS-PLATFORM**
COURSE CODE	**05MB0104**
COURSE CREDITS	**4**

Objective:

1 To develop an understanding of the cross-platform development.
2 To learn about the language used for cross-platform development.
3 To develop their concepts about control and looping statements.
4 To learn about object-oriented approaches in language.
5 To develop an understanding of collection frameworks.

Course Outcomes: After completion of this course, student will be able to:

1 Understand Dart language basics, compare with JavaScript, and set up development environment with Dart SDK and WebStorm.
2 Candidates will able to understand Dart syntax, create and execute Dart programs, work with identifiers, command line options, keywords, comments, data types, variables, and operators.
3 Students will able to Learn control flow structures, and loop control statements, and understand their usage.
4 Students will Master Object-Oriented Programming (OOP) concepts such as classes, objects, functions, interfaces, and generics in Dart, and learn to handle packages and exceptions efficiently.
5 Master Dart's collection framework (lists, sets, maps), debugging, typedef, libraries, async programming, and concurrency.

Pre-requisite of course: Basic Knowledge of Programming

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
2	0	4	50	30	20	0	50

Contents : Unit	Topics	Contact Hours
1	**BASICS TO DART** Introduction of Dart language, features, and the difference between Dart and JavaScript., Environment setup, dart SDK, online execution, IDE for dart, WebStorm configuration., Syntax of the dart, First dart program. , Execution of dart program, identifiers., Dart command line options, keywords, comments., Data types, variables, Operators.	15

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
2	**CONTROLLING, LOOPING & OBJECT-ORIENTED PROGRAMMING** If, if-else, else if ladder, nested if., switch case, break, loop syntax, for loop., while loop, do while loop, labels., continue, break in loop., Understand OOP, Work with Class and Object, Functions, Interfaces, Generics., Packages, Exceptions, Collections-List, Set, Map, , Queue, Enumerations, Debugging, Typedef, Libraries, Async, Concurrency	15
Total Hours		**30**

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Unit 1** Go to DartPad (https://dartpad.dev/) and write a Dart program that calculates the area of a rectangle. Prompt the user to enter the length and width of the rectangle, calculate the area, and print the result., Install Dart SDK on your local machine and verify the installation by running the dart --version command in the terminal/command prompt., Set up a new Dart project using your preferred code editor or IDE (such as Visual Studio Code or IntelliJ IDEA) and create a basic Dart file with a "Hello, World!" print statement. Run the file to ensure everything is set up correctly., Write a Dart program that declares variables of different data types (int, double, String, bool) and initializes them with values. Print out each variable's value., Create a Dart function that takes two parameters (num1 and num2) and returns the sum of the two numbers. Call the function and print the result., Write a Dart program that prints "Hello, World!" to the console., Create a Dart program that calculates the sum of two numbers entered by the user and displays the result., Execute a Dart program using a command-line interface (CLI) and observe the output., Define and use identifiers (variables and functions) with meaningful names in a Dart program to perform basic arithmetic operations., Add single-line and multi-line comments to a Dart program to document code functionality., Define variables of various data types (int, double, String, bool) in a Dart program and initialize them with values., Perform arithmetic operations (addition, subtraction, multiplication, division, modulo) using operators in Dart and display the results., Use relational and logical operators in conditional statements to compare values and control program flow.	30

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
2	**Unit 2** Write a Dart program to determine if a given number is positive, negative, or zero using if-else statements .Implement a Dart program to check if a student's grade falls into different categories (A, B, C, D, F) using if-else if ladder., Write a Dart program to display the name of a month based on its number (1-12) using a switch case statement., Implement a Dart program to print the first 10 even numbers using a for loop., Create a Dart program to calculate the factorial of a number using a for loop. Write a Dart program to display numeric series up to a specified number using a while loop and do while loop., Write a Dart program to print all the even numbers between 1 and 20 except for 10 using the continue statement., Implement a Dart program that models a car object with properties like make, model, and year. Use object-oriented principles to define methods for starting the car, accelerating, and braking., Create a Dart class representing a student with properties like name, age, and grade. Instantiate multiple student objects and demonstrate accessing their properties and methods., Create a Dart package named "calculator" containing classes for basic arithmetic operations (addition, subtraction, multiplication, division). Import this package into a Dart program and perform arithmetic operations using its classes., Implement error handling in a Dart program that divides two numbers. Use a try-catch block to catch and handle exceptions such as division by zero., Create a Dart program that demonstrates the use of lists to store and manipulate a list of integers, strings, or objects. Implement a Dart program that utilizes sets to remove duplicate elements from a list of integers or strings., Write a Dart program to store key-value pairs in a map and perform operations like adding, updating, and removing entries., Implement a Dart program that simulates a queue data structure using lists and demonstrates queue operations such as enqueue, dequeue, and peek., Define a typedef in Dart for a function signature and use it to declare a function variable. Demonstrate passing functions as parameters or returning them from other functions., Create a Dart library containing reusable functions or classes and import it into a program to utilize its functionalities.	30
Total Hours		**60**

Textbook :

1 Beginning Dart: Learn Programming While Applying It On Real Projects, Yury Magalif, Magalif, 2011

References:

1 Dart Apprentice: Fundamentals, Dart Apprentice: Fundamentals, Jonathan Sande, Razeware LLC, 2021

2 Dart Apprentice: Beyond the Basics, Dart Apprentice: Beyond the Basics, Jonathan Sande, Razeware LLC, 2021

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

References:

3 Data Structures & Algorithms in Dart, Data Structures & Algorithms in Dart, Jonathan Sande, Vincent Ngo, and Kelvin Lau, Razeware LLC, 2021

4 O guia de Dart, O guia de Dart, Julio Bitencourt, Casa do Código, 2014

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
Remember / Knowledge	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Higher order Thinking / Creative**
10.00	20.00	25.00	25.00	10.00	10.00

Instructional Method:

1 Board work

2 PPT

3 Demo

Supplementary Resources:

1 https://dart.dev/language

2 https://www.geeksforgeeks.org/dart-tutorial/

3 https://www.javatpoint.com/dart-programming

DR. SUNIL LALCHAND BAJEJA

Digitally signed by (Name of HOD)

DR. R SRIDARAN

Digitally signed by (Name of Dean/ Principal)