Assignment

Mobile Application Design Lab

Course Code: CSE414

Instructor : **Nishat Sadaf Lira**

Name	Faysal Hassan Torjo
ID	221-15-5926
Section	61_T1

Lab Report: Introduction to Dart

Objective:

To understand the basics of Dart programming, including syntax, data structures, object-oriented programming (OOP), and inheritance.

Introduction:

Dart is a client-optimized programming language developed by Google, primarily used for building mobile, web, and desktop applications. Flutter, a UI toolkit, is based on Dart and helps create cross-platform applications.

Basic Dart Syntax:

Dart syntax is similar to Java, C++, and JavaScript. Example:

```
void main() {
 print("Hello, Dart!");
}
```

- void indicates no return value.
- main() is the entry point.
- print() displays output.

Lists in Dart:

Lists are ordered collections that can store multiple elements.

```
List<String> names = ["Alice", "Bob", "Charlie"];
names.add("David");
print(names);
```

Sets in Dart:

Sets store unique elements without duplicates.

```
Set<int> uniqueNumbers = {1, 2, 3, 4, 5};
uniqueNumbers.add(5); // Not added as it's a duplicate
```

```
print(uniqueNumbers);
```

Object-Oriented Programming in Dart:

Dart supports classes and objects.

```
class Student {
 String name;
 int id;
 Student(this.name, this.id);
 void display() {
 print("Student Name: $name, ID: $id");
 }
}
void main() {
 Student student1 = Student("John Doe", 12345);
 student1.display();
}
```

Functions in Dart:

Functions enable code reusability.

```
int addNumbers(int a, int b) {
 return a + b;
}
void main() {
 int sum = addNumbers(5, 10);
 print("Sum: $sum");
}
```

Inheritance in Dart:

A child class can inherit properties and methods from a parent class.

```
class Animal {
 void makeSound() {
 print("Animal makes a sound");
 }
}
class Dog extends Animal {
 @override
 void makeSound() {
 print("Dog barks");
 }
}
void main() {
 Dog dog = Dog();
 dog.makeSound();
}
```

Abstract Classes in Dart:

Abstract classes define interfaces.

```
abstract class Vehicle {
 void move();
}
class Car implements Vehicle {
 @override
 void move() {
 print("Car is moving");
 }
}
void main() {
 Car myCar = Car();
 myCar.move();
}
```

Conclusion:

Dart is a powerful, object-oriented language that is essential for Flutter development. Understanding lists, sets, functions, and OOP concepts is crucial for building robust applications.