Chapter 3

Introduction to Dart Programming

Dart

- Dart is an ***open-source*** general-purpose ***programming language***.
- It is originally developed by **Google**.
- Dart is an ***object-oriented language*** with C-style syntax.
- It supports programming concepts like ***interfaces, classes***, unlike other programming languages ***Dart doesn’t*** support arrays.
- ***Dart collections*** can be used to replicate data structures such as ***arrays, generics, and optional typing***.

The following code shows a simple Dart program

```
void main() {
  print("Dart language is easy to learn");
}
```

Variables and Data types

- ***Variable*** is named ***storage location*** and ***Data types*** simply refers to the **type** and **size** of data associated with **variables** and **functions**.
- Dart uses ***var*** keyword to declare the **variable**.
- The syntax of ***var*** is defined below,

  var name = 'Dart';

- **The final and const keyword are used to declare constants.**
    - **They are defined as below –**

```
void main() {
  final a = 12;
  const pi = 3.14;
  print(a);
  print(pi);
}
```

Dart language supports the following data types

- **Numbers** − It is used to represent ***numeric*** literals – Integer and Double.
- **Strings** − It represents a sequence of characters. **String** values are specified in either **single** or **double quotes**.
- **Booleans** − Dart uses the ***bool*** keyword to represent **Boolean** values – **true** and **false**.
- **Lists and Maps** − It is used to represent a collection of objects.
  - A simple List can be defined as below −.

```
void main() {
  var list = [1,2,3,4,5];
  print(list);
}
```

- The list shown above produces [1,2,3,4,5] list.

Cont..

- **Map can be defined as shown here**

```
void main() {
  var mapping = {'id': 1,'name':'Dart'};
  print(mapping);
}
```

Dynamic :- If the variable type is ***not*** defined, then its default type is dynamic.

- **The following example illustrates the dynamic type variable**

```
void main() {

  dynamic name = "Dart";

  print(name);

}
```

Decision Making and Loops

- A decision making block evaluates a condition before the instructions are **executed**. Dart supports ***If, If..else and switch*** statements.
- Loops are used to repeat a block of code until a specific condition is met.
  - Dart supports for, for..in , while and do..while loops.
  - Let us understand a simple example about the usage of control statements and loops

```
void main() {
  for( var i = 1 ; i <= 10; i++ ) {
    if(i%2==0) {
      print(i);
    }
  }
}
```

The above code prints the ***even*** numbers from ***1 to 10***.

Functions

- A function is a group of **statements** that together performs a specific task. Let us look into a simple function in **Dart** as shown here:-

```
void main() {
  add(3,4);
}
void add(int a,int b) {
  int c;
  c = a+b;
  print(c);
}
```

The above function *adds two* values and ***produces 7*** as the ***output***.

Object Oriented Programming

- Dart is an ***object-oriented*** language. It supports **object-oriented** programming features like classes, interfaces, etc.
- A class is a **blueprint** for creating objects. A class definition includes the following :-
  - Fields
  - Getters and setters
  - Constructors
  - Functions

Now, let us create a simple class using the above definitions:–

```
class Employee {
  String name;

  //getter method
  String get emp_name {
    return name;
  }
  //setter method
  void set emp_name(String name) {
    this.name = name;
  }
  //function definition
  void result() {
    print(name);
  }
}
void main() {
  //object creation
  Employee emp = new Employee();
  emp.name = "employee1";
  emp.result(); //function call
```

THANK YOU

?