Dart Programming Essentials

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

Ans :

Dart is a statically typed language, meaning variables have specific types.

1: int(Integer)

-> Integer is use to store numerical values like

1,2,3,4,5....

-> integer store 64-bit signed values

-> EX: int a = 25;

2: double

-> double is use to store decimal numbers like

4.1564

-> double store 64-bit floating point

-> EX: double d = 3.142..;

**3: String**

**-> String is use to store sequence of characters like**

**"name"**

**-> Can be declared using single ('') or double (" ") quotes.**

**-> EX: String str = "name";**

**4: List(Array)**

**-> A collection of ordered elements like [1,3,4]**

**-> Can store heterogeneous data**

**-> EX: List<String> fruits = ["Apple", "Banana", "Cherry"];**

**5: Map**

**I**

**-> Dictionary-like data structure**

**-> Stores values mapped to unique keys. Keys can be any data type (commonly String or int)**

**-> EX: Map stu = {"name": 1,**

**"name2": 2};**

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

Ans :

Control structures in Dart help manage the flow of execution of a program by determining the conditions under which certain blocks of code are executed.

-> if-else if-else:

void main() {

int marks = 85;

if (marks >= 90) {

print("Grade: A");

} else if (marks >= 75) {

print("Grade: B");

} else {

print("Grade: C");

}

}

-> Looping Statements:

a) for Loop:

void main() {

for (int i = 1; i <= 5; i++) {

print("Iteration $i");

}

}

-> for-in Loop:

void main() {

List<String> fruits = ["Apple", "Banana", "Cherry"];

for (String fruit in fruits) {

print(fruit);

}

}

-> while Loop:

void main() {

int count = 1;

while (count <= 5) {

print("Count: $count");

count++;

}

}

-> Switch-Case Statement:

void main() {

String grade = "B";

switch (grade) {

case "A":

print("Excellent!");

break;

case "B":

print("Good Job!");

break;

case "C":

print("Needs Improvement.");

break;

default:

print("Invalid Grade.");

  }

}

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

Ans :

-> Classes and Objects

A class is a blueprint for creating objects. It defines properties (variables) and behaviors (methods).

An object is an instance of a class that holds actual values and can perform actions defined in the class.

Example: A Car class can have properties like brand and speed, and objects like Tesla and BMW can be created from it.

-> Inheritance

Inheritance allows a class (child class) to inherit properties and methods from another class (parent class).

This promotes code reusability, as the child class can use the existing functionalities of the parent class instead of rewriting them.

Example: A Dog class can inherit from an Animal class and get its general behaviors while adding specific behaviors like barking.

-> Polymorphism

Polymorphism means "many forms" and allows a method to have different behaviors in different classes.

This is mainly done through method overriding, where a child class redefines a method from the parent class to give it a different functionality.

Ex : A Bird class might have a method makeSound(), which different birds can override to produce different sounds.

-> Abstract Classes

An abstract class is a class that cannot be instantiated (cannot create an object of it).

It serves as a template for other classes and often contains methods that must be implemented by subclasses.

Ex : A Shape class can be abstract, defining a method draw() that different shapes (like Circle and Rectangle) must implement.

-> Interfaces

Dart does not have a separate interface keyword; instead, any class can act as an interface if another class implements it.

Interfaces define a contract that a class must follow by implementing all the methods.

Ex : A Printable interface may require classes like Document and Image to implement a printData() method.