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

1. **Int: it represents the numbers (positive, negative, zero)**

**Use: - it’s used for counting, indexing, and other integer-based values**

**Ex: - int age = 30;**

1. **Double: it represents the number’s with decimal values.**

**Use: - it’s used for prices values like price height and percentage etc…**

**Ex: - double price = 19.99;**

1. **String: - it represents a sequence of character.**

**Use: it used for names, words messages or any word text data.**

**Ex: String name = "ABC XYZ ";**

1. **List: -** **A List is an ordered collection of objects, similar to an array. Elements are accessed by a zero-based index**.

**Use: - it’s used for storing multiple items in sequence.**

**Example: List<String> fruits = ["apple", "banana", "orange"];**

1. **Map: - A Map is a collection of key-value pairs, where each key must be unique.**

**Use: - it’s used for when data needs to be stored and retrieved by a key.**

**Example: - Map<String, int> scores = {"Alice": 95, "Bob": 88};**

1. **Set: - its removed duplicates values.**

**Use: - when you want only distinct elements.**

**Example: - var numbers = {1, 2, 3, 3, 4};**

**// Set automatically removes duplicate 3 → {1, 2, 3, 4}**

1. **Boolean: - it represents only two vales**

**True or false**

**Use: - conditions, comparisons**, **and logical operators**

**Example:- bool is Logged IN = true; // stores a true/false value**

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

1. **If statements: - the if statements are used to execute a block of code only if the condition evaluates to true.**

**Ex:**

**Void main ()**

**{**

**Int age = 20;**

**If (age >=18)**

**{**

**Print (“Eligible for vote”);**

**}**

**}**

1. **Is-else: - the is else statement allows you to execute one block of code if the conditions are true and another block if condition is false.**

**Ex:**

**Void main ()**

**{**

**Int marks > 35;**

**If (marks > 40)**

**{**

**Print(“pass”);**

**}**

**Else**

**{**

**Print (“fail”);**

**}**

**}**

1. **For Loop: - for loop is control structure that repeats code a fixed number of times**

**Ex: -**

**void main ()**

**{**

**for (int I = 1; I <= 5; I++)**

**{**

**print ("Count: $I");**

**}**

**}**

1. **While: - the while loop keeps executing a block of code as long as conditions is true**

**This is only an entry-controlled loop**

**Ex:**

**void main ()**

**{**

**int I = 1;**

**while (I <= 5)**

**{**

**print ("Number: $I");**

**I++;**

**}**

**}**

1. **Switch – case: - A switch case statement allows execution of one block of code from multiple choice based on the value of an expression**

**EX: -**

**void main ()**

**{**

**String grade = "B";**

**switch (grade)**

**{**

**case "A":**

**print("Excellent");**

**break;**

**case "B":**

**print("Good");**

**break;**

**case "C":**

**print("Average");**

**break;**

**default:**

**print ("Invalid grade");**

**}**

**}**

1. **Do-while:** - **This is an** **exit-controlled loop** **(condition checked later).**

**EX:** -

**void main ()**

**{**

**int I = 1;**

**do {**

**print ("Value: $I");**

**if++;**

**}**

**while (I <= 5);**

**}**

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

1. **Class: group of data members & member functions Like person can be class having data members height and weight and member functions as get details () and put details () to manipulate on details**

**Ex:**

**class Mobile**

**{**

**String brand;**

**Mobile (this. Brand);**

**void call () => print ("$brand is making a call");**

**}**

**void main () {**

**Mobile m = Mobile("iPhone");**

**m.call (); // iPhone is making a call**

**}**

1. **Inheritance: - When One Object Acquire All the Properties and Behaviour of Parent class That is called an Inheritance.**

**Ex:**

**class Mobile {**

**void call () => print ("Calling...");**

**}**

**class Smartphone extends Mobile {**

**void browse () => print ("Browsing Internet");**

**}**

**void main () {**

**Smartphone s = Smartphone ();**

**s. call (); // from parent**

**s. browse (); // from child**

**}**

1. **Polymorphism: - many ways to perform anything**
   * 1. **Method overloading: dart not support overloading**
     2. **Method overriding: When we declare the same method in the subclass, which is previously defined in the superclass is known as the method overriding. The subclass can define the same method by providing its own implementation, which is already exists in the superclass. The method in the superclass is called method overridden, and method in the subclass is called method overriding.**
2. **Interface: An interface defines the syntax that any entity must adhere to. Dart does not have any separate syntax to define interfaces. An Interface defines the same as the class where any set of methods can be accessed by an object. The Class declaration can interface itself**

* **Describe asynchronous programming in Dart, including Future, async, await, and Stream**.

1. **Async: It simply allows us to write promises-based code as if it was synchronous and it checks that we are not breaking the execution thread. It operates asynchronously via the event-loop. Async functions will always return a value. It makes sure that a promise is returned and if it is not returned then JavaScript automatically wraps it in a promise which is resolved with its value**
2. **Await: Await function is used to wait for the promise. It could be used within the async block only. It makes the code wait until the promise returns a result. It only makes the async block wait**
3. **Stream: - A Stream represents a sequence of asynchronous data events**. **Unlike a Future (which gives one value), a Stream can deliver multiple values over time. You can** **listen to a stream and react whenever a new value comes. Streams are useful for real-time data (sensors, chat messages, stock prices, location tracking).**
4. **Future:** **A Future represents a result that will be available in the future**. **It is** **single-shot: it completes once**.