Module : 2

Dart Programming Essentials

1. Explains the fundamental data types in Dart (int, double, string, list, Map etc...) and their uses.

→ int :- Whole numbers (without decimal point).

Example :- 10, -5, 100.

Use :- When we want to store numbers like age, quantity, marks etc.

```
int marks = 20;
```

→ double :- Decimal numbers (number with a point)

Example :- 3.14, -0.5,

Use :- For values like price, height, weight etc..

```
double h = 3.25;
```

→ String :- A sequence of characters (text).

Example :- "abc", "Welcome"

Use :- To store names, msg or any kind of text.

→ List :- Collection of items that can hold multiple values of same data types.

Example :- [1, 2, 4, 5]

Use :- Lists are used to store sequence of data.

→ Map :- A collection of key-value pair

Example Map <> a = {"abc" : 1};

Use :- Maps are used to associate arrays or dictionaries.

→ Set :- Represent a collection of unique items.

Example :- Set <String> fruit = {"Apple, Banana"};

Use :- Sets are used when you wants to ensure that no duplicates are present.

2. Describe control structures in Dart with examples.

⇒ Control structure allows you to control the flow of execution in Dart Programs.

→ Here are some common control structures:

- if :-

  Execute a block of code if the condition is true.

Example :-

```
if (age >= 18)
{
    print (" You are adult ");
}
```

- else :-

  Execute a block of code if the condition in if statement is false.

Example :-

```
if (age >= 18)
{
    print (" adult ");
}
else {
```

```
    print("minor");
}
```

- for loop:-
  A loop ~~spec~~ if iterates a specific number of times.

Example:

```
for(int i = 0 ; i<= 5 ; i++)
{ print(i); }
```

- while:-
  A loop continues as long as a condition is true.

Example:

```
int c = 0
while ( c <= 5)
{
    print(c);
    c++;
}
```

- switch:-
  A control statement that allows you to execute different block of code based on value of a variable

Example:

```
String day = "Monday";
switch (day)
{
  case "Monday":
    print("Start of the week");
    break;
  case "Friday":
    print("End of the week");
    break;
  default:
    print("Midweek");
}
```

3. Explain object oriented programming concepts in Dart.

⇒ Dart is a object oriented programming language, which means it uses objects and classes to organize a code.

→ Class:-

- Class is a blueprint of for creating objects.
- A class contains fields (attributes) and methods (functions).

→ Inheritance:-

- A mechanism where new class ca inherit properties and methods fro existing class.

→ Polymorphism:

- The ability to call the same method on different objects and have each respond in their own way.

→ Object:

- Any entity which has state and behaviour is called an object.

→ ~~Encapsulation~~:- Abstraction

- Hiding internals details and showing only functionalities.

→ Encapsulation:-

- Wrapping up of data in a single unit.

4. Describe ~~asy~~ asynchronous programming in Dart, including Future, async, wait, and Stream.

⇒ Dart provides a support for asynchronous programming, which allows you to perform tasks without blocking the main thread.

→ Future :-

Represents a value that may be available at some point in future.

→ async :-

A keyword that allows you to define an ~~an~~ asynchronous function. It enables the use of "await" keyword within functio

→ await :-

A keyword that pause the execution of an async function until the future completes.