NAME : ANKIT DUBEY ROLL NO : 28 T12

Lab Assignment 1

AIM: Study and implement basic Dart programs using loops and function.

LO1: Understand cross platform mobile application development using the flutter framework.

THEORY:

Dart is a programming language developed by Google, primarily for building mobile, desktop, server, and web applications. It is designed with a focus on simplicity, performance, and productivity. Dart is often associated with the Flutter framework, which is a UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase.

Basic Dart Program Structure:

A basic Dart program typically consists of the following components:

1. Main Function (main()) - Entry Point:

Every Dart program starts its execution from the main function.

The main function is the entry point and is required in every Dart program.

2. Comments:

Dart supports both single-line (//) and multi-line (/* */) comments.

Comments are used to add explanations or notes to the code.

3. Print Statements:

print() function is used to output text to the console.

It is commonly used for debugging and displaying information.

Datatypes in Dart

In Dart, data types are used to specify the nature of values that variables can hold. Dart is a statically-typed language, meaning the data type of a variable is explicitly declared at compile time. Here's an overview of common data types in Dart:

int: Represents integer values (whole numbers). double:

Represents floating-point values (decimal numbers).

String: Represents sequences of characters, used for handling textual data. bool:

Represents boolean values, true or false.

List: Represents an ordered collection of elements. Lists can contain values of different data types.

Map: Represents a collection of key-value pairs, where each key is associated with a value.

dynamic: Represents a variable with a dynamic type. The type of a dynamic variable can change during runtime.

Object: The root type for most types in Dart. All classes inherit from the Object class.

Function: Represents a function type.

void: Represents the absence of a value. Used as a return type for functions that do not return any value.

Loop Structures in Dart:

1. for loop:

The for loop is used for iterating over a range of values.

It typically consists of an initialization statement, a condition for iteration, and an update statement.

2. while loop:

The while loop repeats a block of code while a given condition is true.

It checks the condition before each iteration.

3. do-while loop:

The do-while loop is similar to the while loop, but it checks the condition after each iteration.

This guarantees that the block of code is executed at least once.

Conditional Structures in Dart:

1. if statement:

The if statement is used to conditionally execute a block of code based on a boolean expression.

It can be followed by an optional else clause to handle the case when the condition is not true.

2. else if statement:

The else if statement allows checking multiple conditions in a sequence.

It is used when there are more than two possible outcomes.

3. else statement:

The else statement is executed when the condition in the if statement is not true.

It provides an alternative block of code to be executed.

4. switch statement:

The switch statement is used for multiple branches of code based on the value of an expression.

It allows comparing a value against multiple possible constant expressions.

PROGRAMS:

1. WAP to read a list of integers and print only even integers.

2. WAP to accept an integer number and print it in words.

NAME : ANKIT DUBEY ROLL NO : 28 T12

DartPad New Samples

```dart

void main() {

  Map<String,String> num = {
    "0":"Zero",
    "1":"One",
    "2":"Two",
    "3":"Three",
    "4":"Four",
    "5":"Five",
    "6":"Six",
    "7":"Seven",
    "8":"Eight",
    "9":"Nine"
  };

  int num1 = 7;
  String snum = num1.toString();

  for(var i =0; i< snum.length;i++){
    print(num[snum[i]]);
  }

}
```

Run

```
Seven
```

3. WAP to input a number and calculate its GCD and LCM using functions.

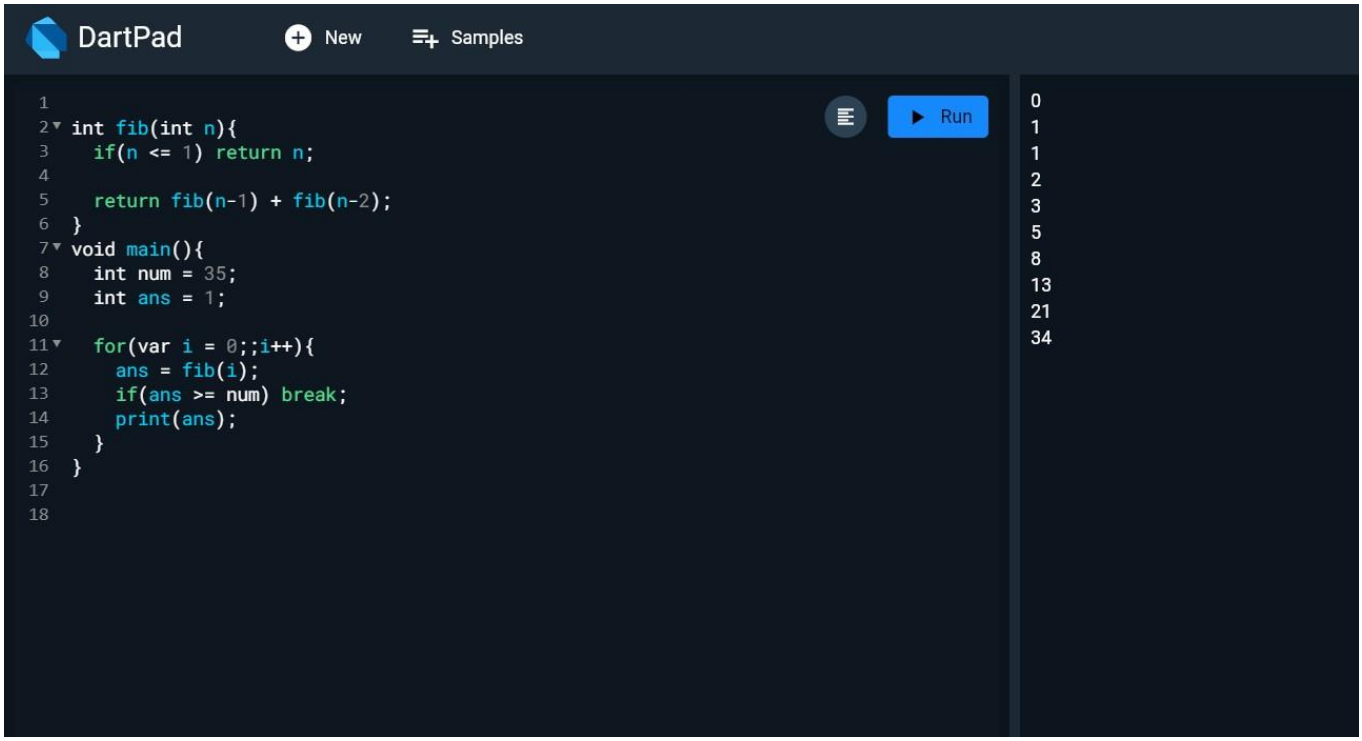
DartPad New Samples

```dart
import "dart:math";

int gcd(int a, int b)
{
  if (a == 0){
   return b;
  }

   return gcd(b % a, a);
}

int lcm(int a,int b){

  int greater = max(a,b);
  int smaller = min(a,b);

  for(var i = greater;; i+=greater){
     if(i%smaller == 0){
       return i;
     }
  }
}

void main() {
  int num1 = 16;
  int num2 = 20;

  int gcdAns = gcd(num1,num2);
  print("GCD of $num1 and $num2 is $gcdAns");

  int lcmAns = lcm(num1,num2);
  print("LCM of $num1 and $num2 is $lcmAns");
}
```

Run

```
GCD of 16 and 20 is 4
LCM of 16 and 20 is 80
```

4. WAP to print Fibonacci series until a given number using functions.



CONCLUSION:

Here we learnt about the basics of dart programming and implemented the programs using dart.