**Week 3:**

**Create an application to implement Decision making and loops using Dart.**

## Decision Making and Loops

The **decision-making** is a feature that allows you to evaluate a condition before the instructions are executed. The Dart language supports the following types of decision-making statements:

* + If statement
  + If-else statement
  + Switch statement

**Loops** are used to execute a block of code repeatedly until a specified condition becomes true. Dart language supports the following types of loop statements:

* for
* for..in
* while
* do..while
  + 1. **If..else statement**

import 'dart:io';

**void main()**

{

print("Enter your favourite number:");

int? dayOfWeek = int.parse(stdin.readLineSync()!);

if (dayOfWeek == 1)

{

print("Day is Sunday.");

}

else if (dayOfWeek == 2)

{

print("Day is Monday.");

}

else if (dayOfWeek == 3)

{

print("Day is Tuesday.");

}

else if (dayOfWeek == 4)

{

print("Day is Wednesday.");

}

else if (dayOfWeek == 5)

{

print("Day is Thursday.");

}

else if (dayOfWeek == 6)

{

print("Day is Friday.");

}

else if (dayOfWeek == 7)

{

print("Day is Saturday.");

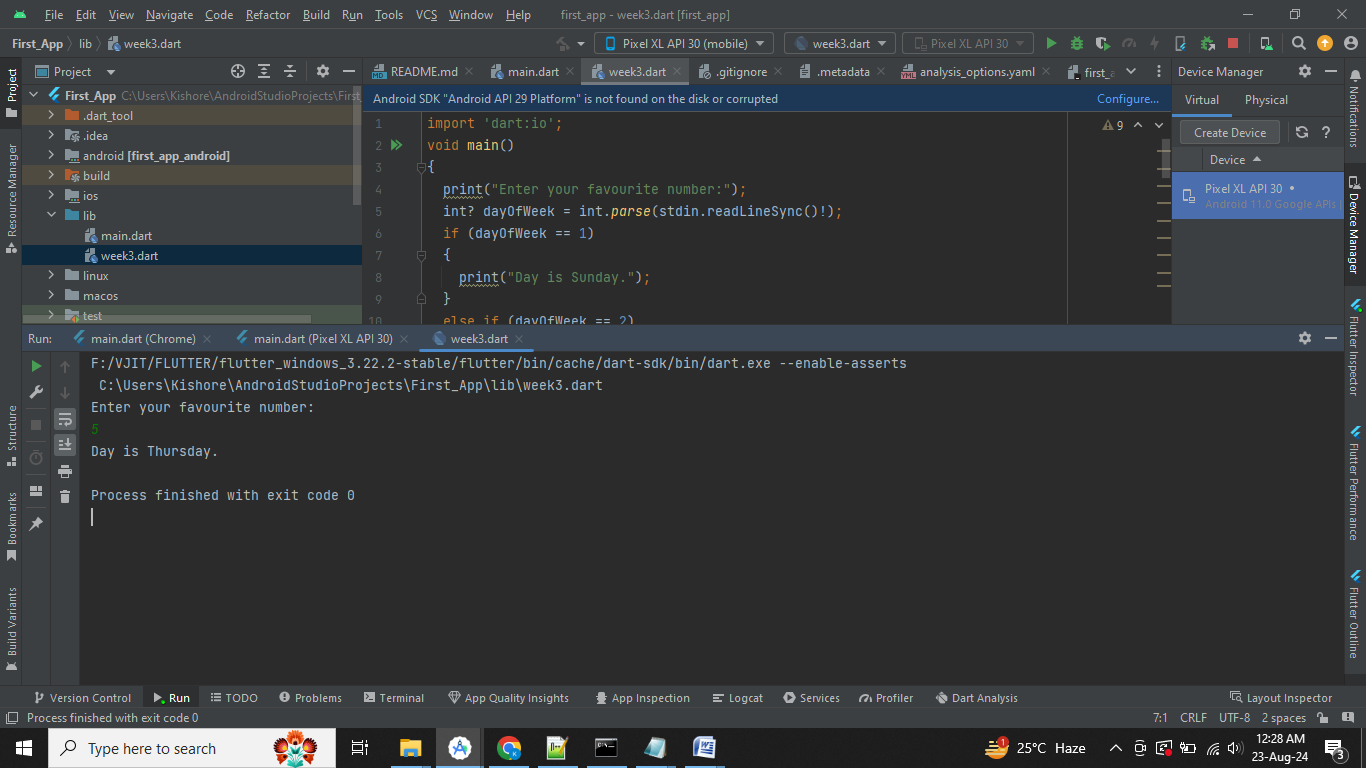
}else

{

print("Invalid Weekday.");

}

}



* + 1. **Switch case example**

import 'dart:io';

**void main()**

{

print("Enter your favourite number:");

int? dayOfWeek = int.parse(stdin.readLineSync()!);

switch (dayOfWeek)

{

case 1:

print("Day is Sunday.");

break;

case 2:

print("Day is Monday.");

break;

case 3:

print("Day is Tuesday.");

break;

case 4:

print("Day is Wednesday.");

break;

case 5:

print("Day is Thursday.");

break;

case 6:

print("Day is Friday.");

break;

case 7:

print("Day is Saturday.");

break;

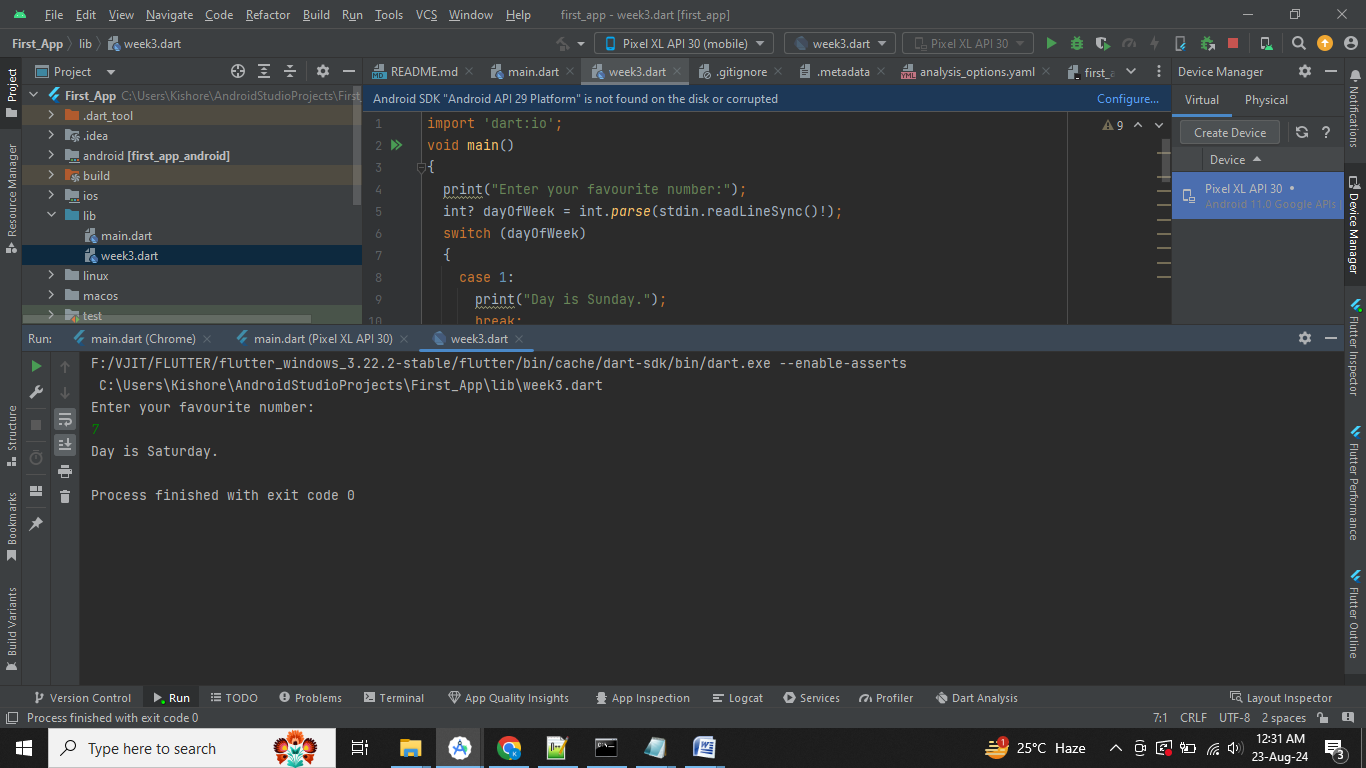
default:

print("Out of range");

break;

}

}



**Loops**

* 1. **For loop**

import 'dart:io';

**void main()**

{

print("Enter your favourite number:");

int? n = int.parse(stdin.readLineSync()!);

print("Multiplication Table of: $n");

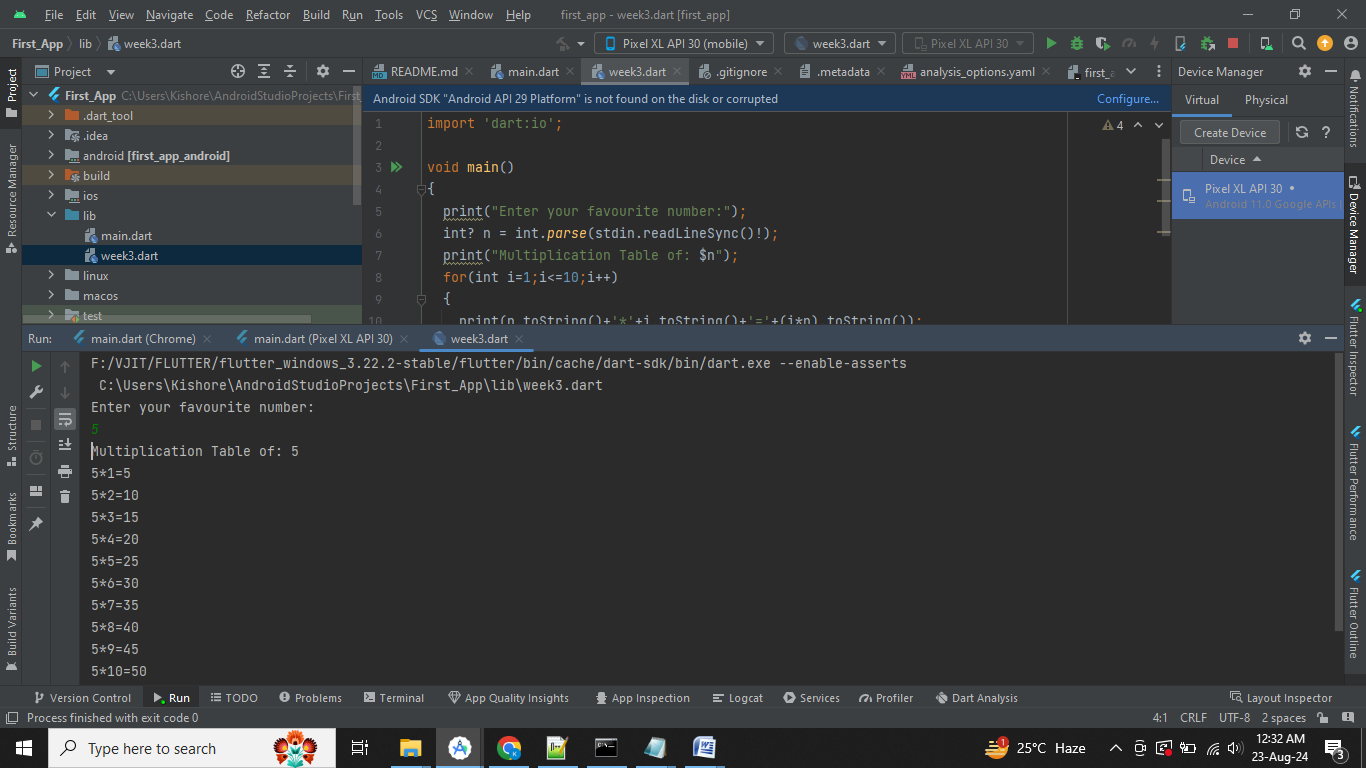
for(int i=1;i<=10;i++)

{

print(n.toString()+'\*'+i.toString()+'='+(i\*n).toString());

}

}

****

1. **For each loop**

**void main()**

{

var list1 = [10,20,30,40,50];

// create an integer variable

int sum = 0;

print("Dart for..in loop Example");

for(var i in list1)

{

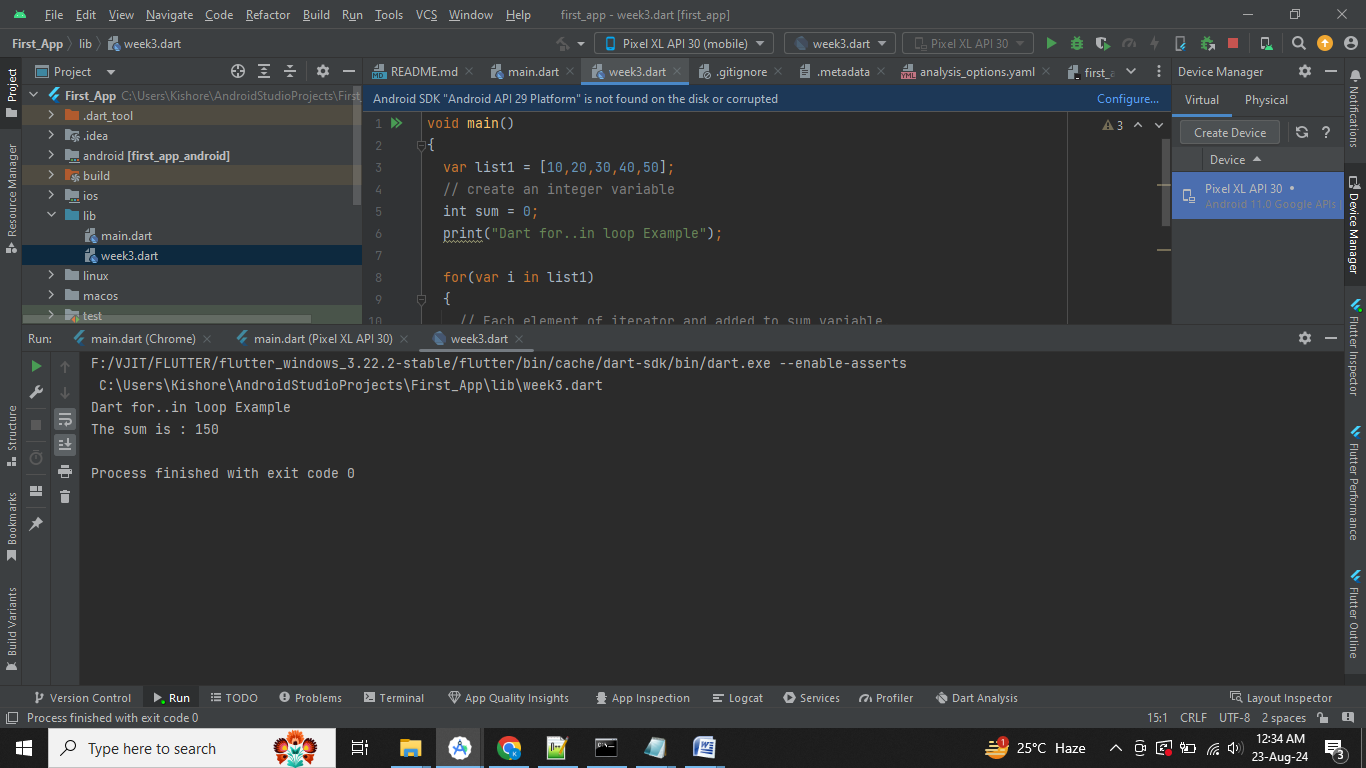
// Each element of iterator and added to sum variable.

sum = i+ sum;

}

print("The sum is : ${sum}");

}



**3) While loop**

**void main()**

{

int total = 0;

int n = 10; // change as per required

int i =1;

print("Display sum of $n natural numbers");

while(i<=n)

{

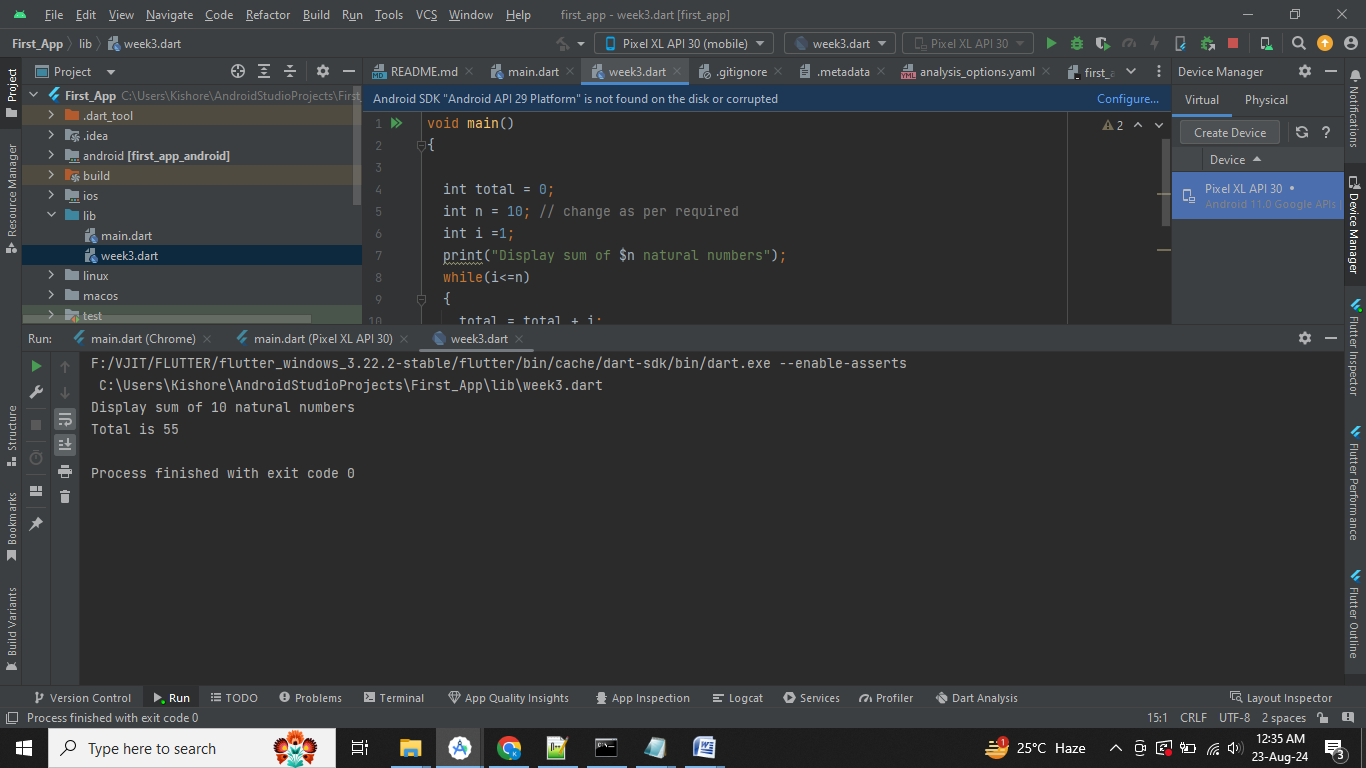
total = total + i;

i++;

}

print("Total is $total");

}



**4) Do..while loop**

**void main()**

{

int total = 0;

int n = 10; // change as per required

int i =1;

print("Display sum of $n natural numbers");

do

{

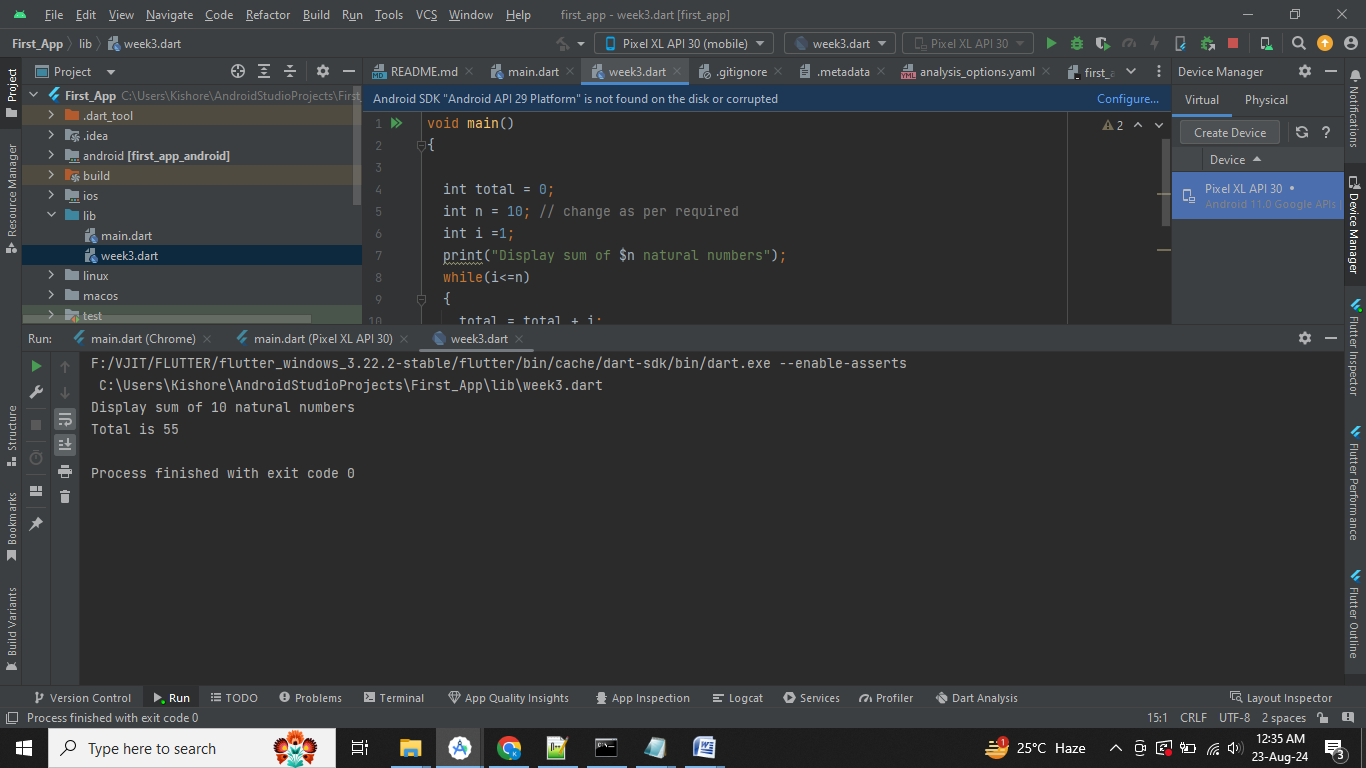
total = total + i;

i++;

}while(i<=n);

print("Total is $total");

}



**Week 4:**

**Create an application to demonstrate user defined functions using Dart.**

**Function Example**

import 'dart:io';

// Creating a Function

**int sum(int a, int b)**

{

// function Body

int result;

result = a+b;

return result;

}

**void main()**

{

print("Example of add two number using the function");

print("Enter your n1:");

int? n1 = int.parse(stdin.readLineSync()!);

print("Enter your n2:");

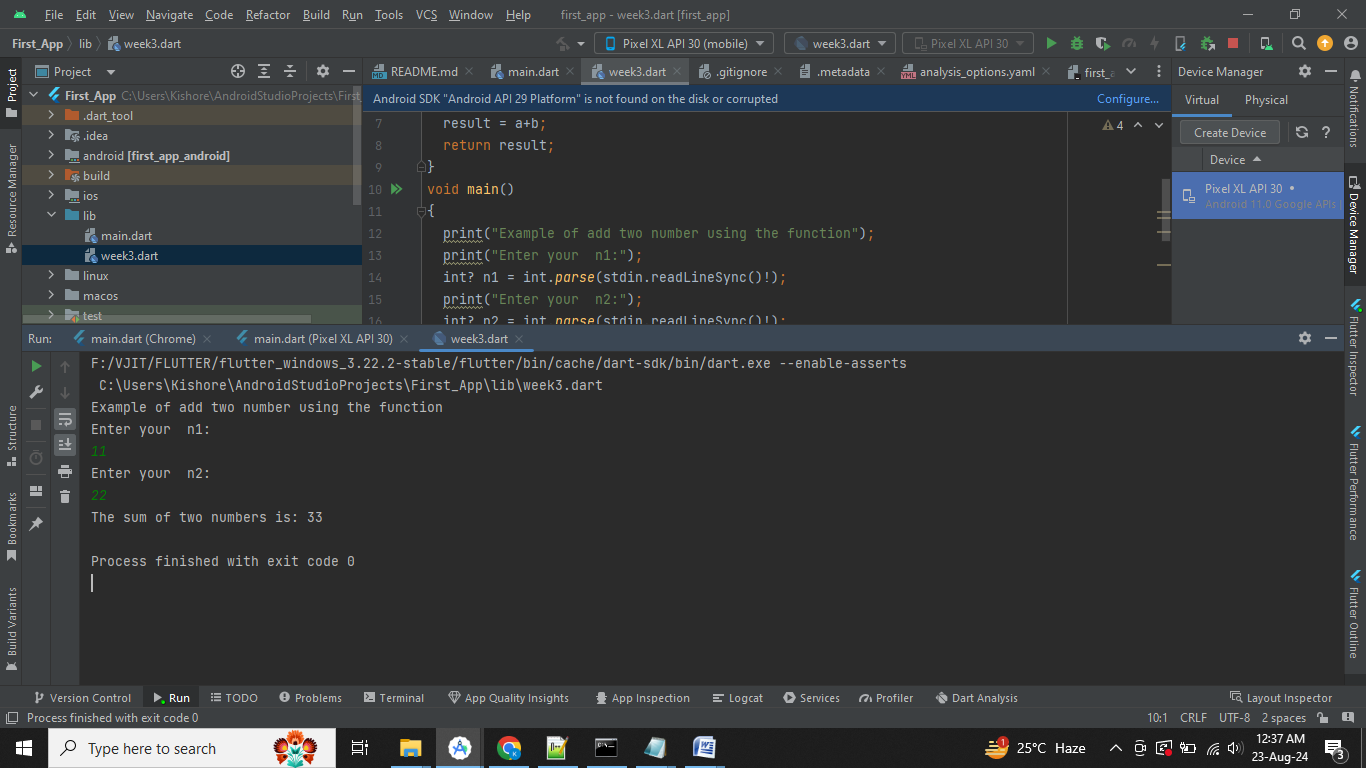
int? n2 = int.parse(stdin.readLineSync()!);

// We are calling a function and storing a result in variable c

var c = sum(n1,n2);

print("The sum of two numbers is: $c");

}



**Recursive Function Example**

import 'dart:io';

**void main()**

{

print("Example of Recursive function");

print("Enter any number:");

int? n = int.parse(stdin.readLineSync()!);

var c=factorial(n);

print("Factorial of a number $n is : $c");

}

**factorial(number)**

{

if (number <= 0)

{

// termination case

return 1;

}

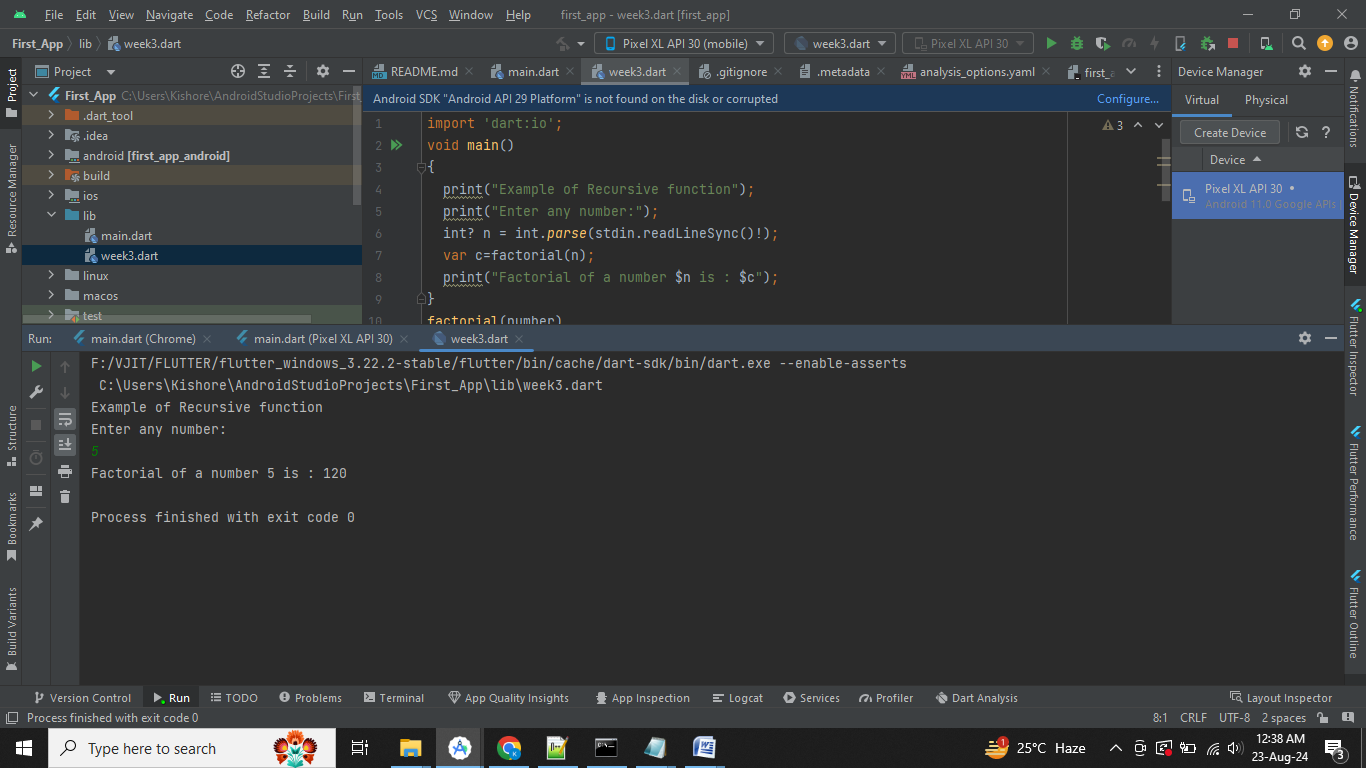
else

{

return (number \* factorial(number - 1)); // function calls itself

}

}

****

**Week 5:**

**Create an application to implement object oriented programming using Dart**

**//Constuctors Example**

**class Student**

{

String? name;

int? age;

int? rollNumber;

// Constructor

Student(String name, int age, int rollNumber)

{

// this is for checking the constructor is called or not.

print( "Constructor called");

this.name = name;

this.age = age;

this.rollNumber = rollNumber;

}

}

**void main()**

{

// Here student is object of class Student.

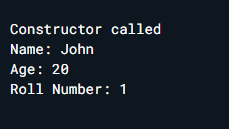
Student student = Student("John", 20, 1);

print("Name: ${student.name}");

print("Age: ${student.age}");

print("Roll Number: ${student.rollNumber}");

}



**Inheritance example**

**class Laptop**

{

// Constructor

Laptop(String name, String color)

{

print("Laptop constructor");

print("Name: $name");

print("Color: $color");

}

}

**class MacBook extends Laptop**

{

// Constructor

MacBook(String name, String color) : super(name, color)

{

print("MacBook constructor");

}

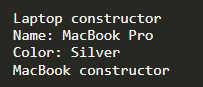
}

**void main()**

{

var macbook = MacBook("MacBook Pro", "Silver");

}



**Interface Example**

import 'dart:io';

// abstract class as interface

**abstract class Person**

{

// properties

String? name;

// abstract method

void run();

void walk();

}

**class Student implements Person**

{

// properties

String? name;

// implementation of run()

@override

void run()

{

print('Student is running');

}

// implementation of walk()

@override

void walk()

{

print('Student is walking');

}

}

**void main()**

{

var student = Student();

print("Enter your name?");

// Reading name from keyboard

String? name = stdin.readLineSync(); // null safety in name string

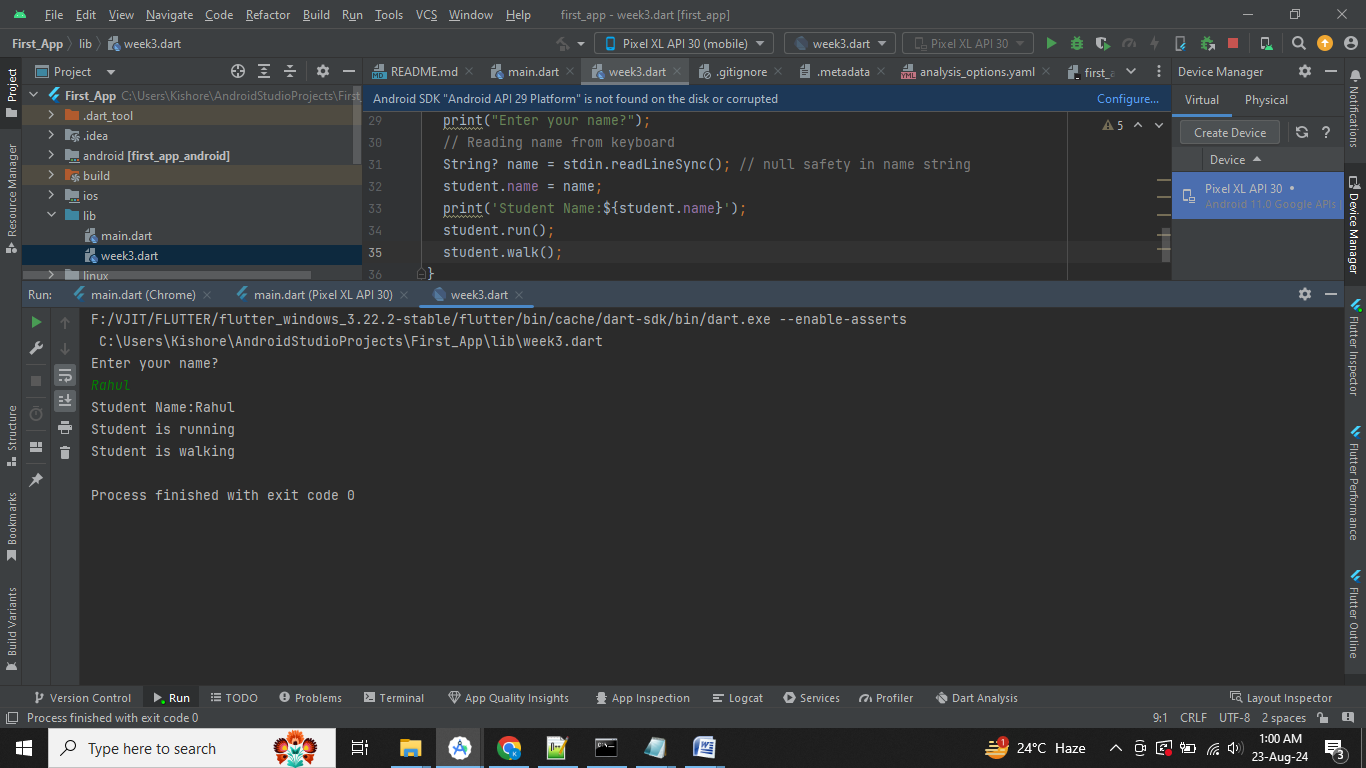
student.name = name;

print('Student Name:${student.name}');

student.run();

student.walk();

}



**Multiple Interfaces Example**

// abstract class as interface

**abstract class CalculateTotal**

{

int total();

}

// abstract class as interface

**abstract class CalculateAverage**

{

double average();

}

// implements multiple interfaces

**class Student implements CalculateTotal, CalculateAverage**

{

// properties

int marks1, marks2, marks3;

// constructor

Student(this.marks1, this.marks2, this.marks3);

// implementation of average()

@override

**double average()**

{

return total() / 3;

}

// implementation of total()

@override

**int total()**

{

return marks1 + marks2 + marks3;

}

}

**void main()**

{

Student student = Student(90, 80, 70);

print('Total marks: ${student.total()}');

print('Average marks: ${student.average()}');

}

