Exception Handling Fundamentals

MODULE 5 CHAPTER 2

Introduction

What Are Exceptions?

- An exception is an **unexpected problem** that stops normal program flow.
 Example: wrong user input, file not found, no internet, etc.

- Exception handling helps the program **continue running without crashing**.

- It makes applications more **stable and user-friendly**.

Why is Exception Handling Important?

- Prevents the program from stopping suddenly.
- Shows clear error messages instead of crashing.
- Helps developers find and fix problems easily.
- Makes your program reliable.

Fundamentals – Terminology & Types

Key Terms

try → code that may cause an error

catch → code that handles the error

finally → code that runs whether there is an error or not

throw → manually create an error

throws → tells the method that it may cause an error

Types of Exceptions

1. **Checked Exception**
   - ○ Must be handled using try-catch.
2. **Unchecked Exception**
   - ○ Happens at runtime.
   - ○ Example: divide by zero, null pointer.

Handling Exceptions

Specific vs General Exceptions

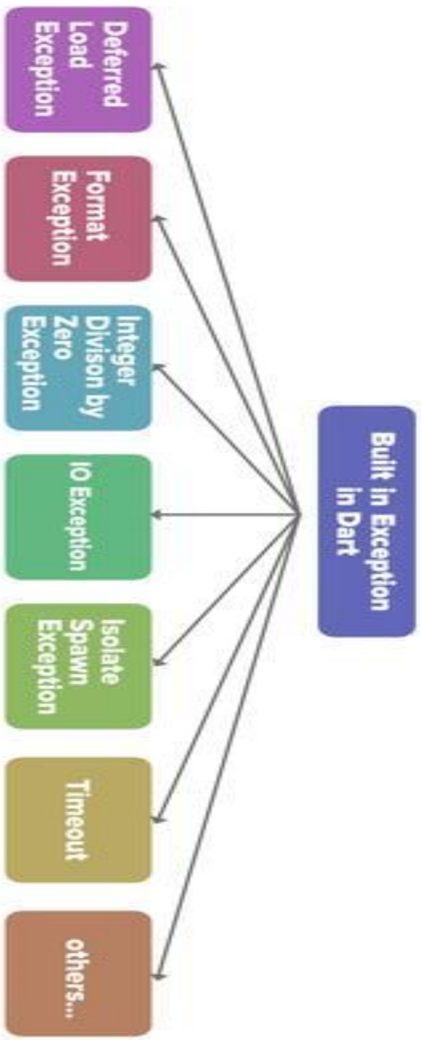
- **Specific exception**: Handles one particular known error. Example: catching only FileNotFoundException.
- **General exception**: Catches any type of error using `Exception`.

Custom Exceptions

- You can create your own exception class.
- Useful when you need special error messages.

Rethrowing Exceptions

- A caught exception can be thrown again.
- Useful when the **main part of the program** should handle it.



Advanced Techniques

Nested try-catch

- Putting one try-catch inside another.
- Useful when different parts of the code need different error handling.

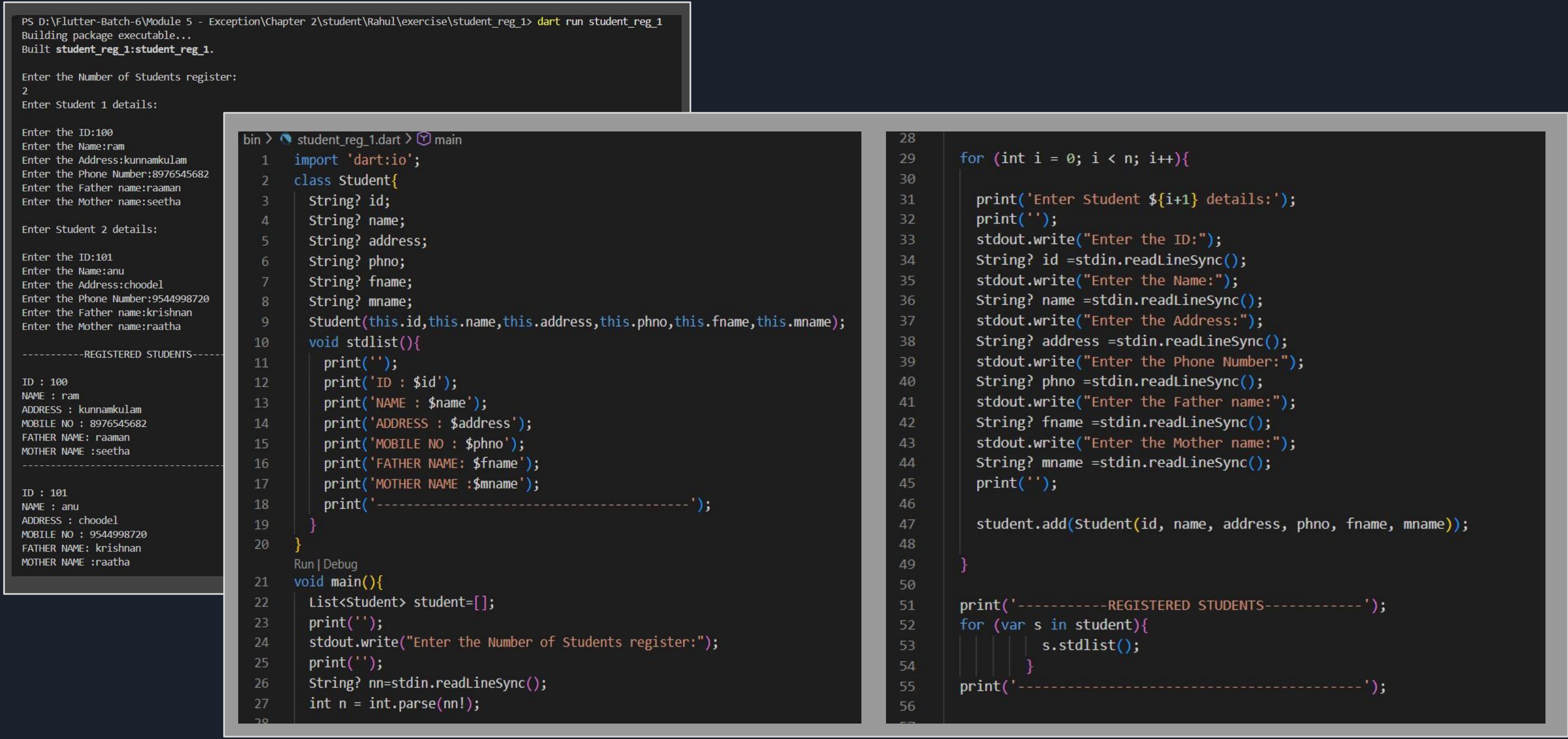
Chained Exceptions

- One exception contains another exception’s information.
- Helps understand the **root cause** of the problem.

Exception Handling in Multithreading

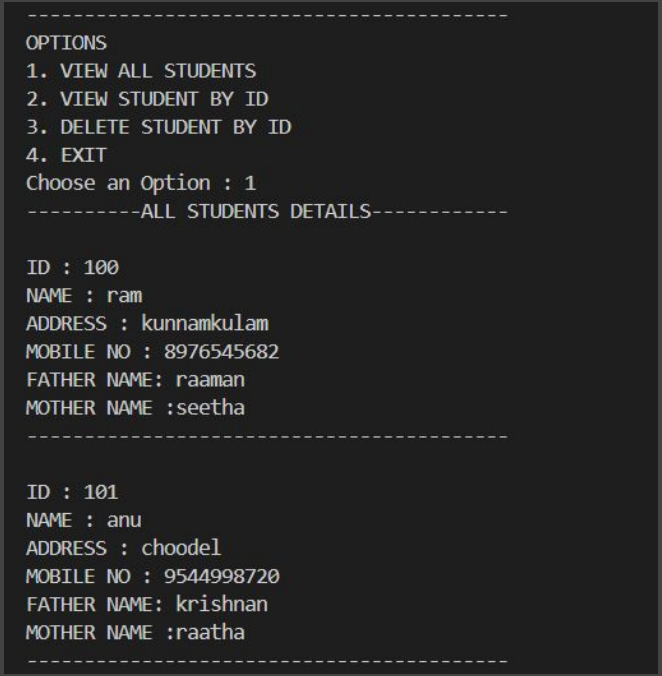
- When multiple threads run, they may try to access the same resource.
- You must use **synchronization** to avoid errors.
- Handle exceptions carefully to make threads safe.

Exercise 1

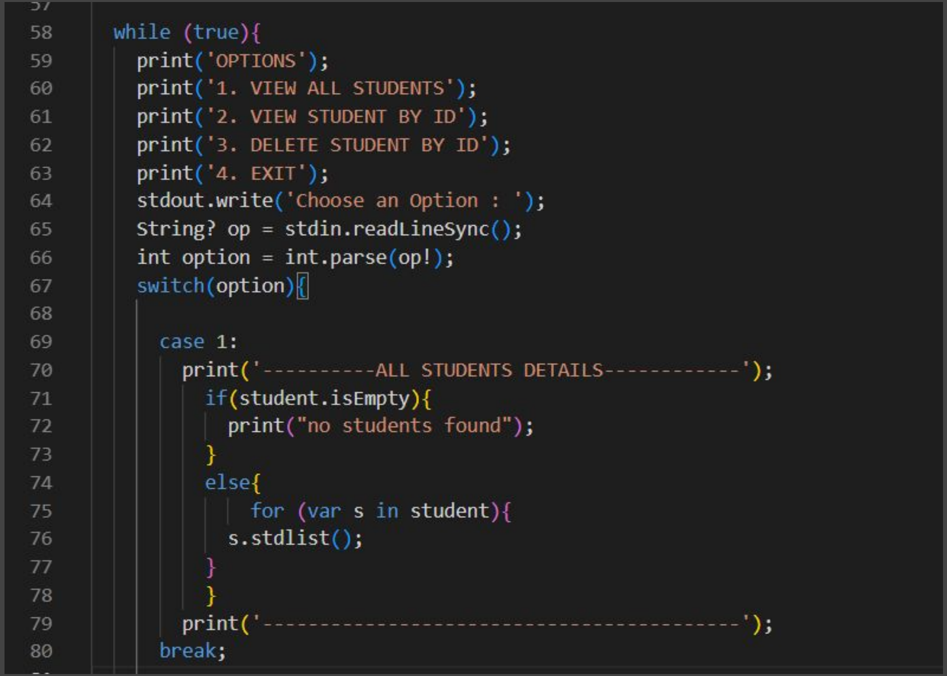


```
PS D:\Flutter-Batch-6\Module 5 - Exception\Chapter 2\student\Rahul\exercise\student_reg_1> dart run student_reg_1
Building package executable...
Built student_reg_1:student_reg_1.

Enter the Number of Students register:
2
Enter Student 1 details:

Enter the ID:100
Enter the Name:ram
Enter the Address:kunnamkulam
Enter the Phone Number:8976545682
Enter the Father name:raaman
Enter the Mother name:seetha

Enter Student 2 details:

Enter the ID:101
Enter the Name:anu
Enter the Address:choodel
Enter the Phone Number:9544998720
Enter the Father name:krishnan
Enter the Mother name:raatha

----------REGISTERED STUDENTS-----

ID : 100
NAME : ram
ADDRESS : kunnamkulam
MOBILE NO : 8976545682
FATHER NAME: raaman
MOTHER NAME :seetha
--------------------------------

ID : 101
NAME : anu
ADDRESS : choodel
MOBILE NO : 9544998720
FATHER NAME: krishnan
MOTHER NAME :raatha
```

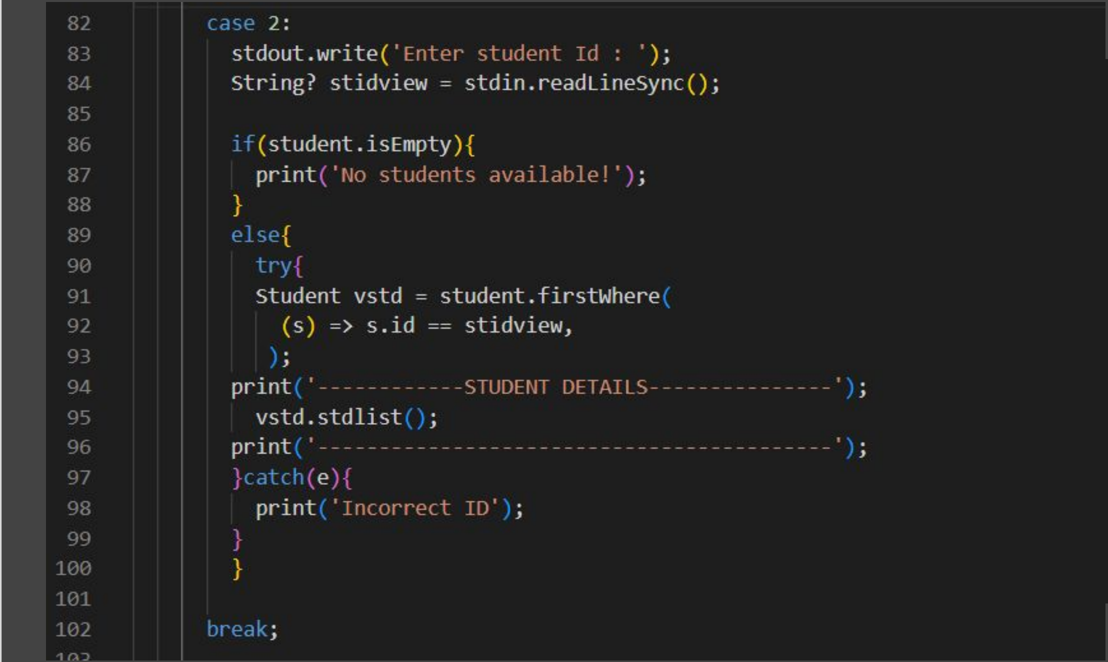
```dart
import 'dart:io';
class Student{
  String? id;
  String? name;
  String? address;
  String? phno;
  String? fname;
  String? mname;
  Student(this.id,this.name,this.address,this.phno,this.fname,this.mname);
  void stdlist(){
    print('');
    print('ID : $id');
    print('NAME : $name');
    print('ADDRESS : $address');
    print('MOBILE NO : $phno');
    print('FATHER NAME: $fname');
    print('MOTHER NAME :$mname');
    print('-----------------------------------------');
  }
}
void main(){
  List<Student> student=[];
  print('');
  stdout.write("Enter the Number of Students register:");
  print('');
  String? nn=stdin.readLineSync();
  int n = int.parse(nn!);

  for (int i = 0; i < n; i++){

    print('Enter Student ${i+1} details:');
    print('');
    stdout.write("Enter the ID:");
    String? id =stdin.readLineSync();
    stdout.write("Enter the Name:");
    String? name =stdin.readLineSync();
    stdout.write("Enter the Address:");
    String? address =stdin.readLineSync();
    stdout.write("Enter the Phone Number:");
    String? phno =stdin.readLineSync();
    stdout.write("Enter the Father name:");
    String? fname =stdin.readLineSync();
    stdout.write("Enter the Mother name:");
    String? mname =stdin.readLineSync();
    print('');

    student.add(Student(id, name, address, phno, fname, mname));

  }

  print('-----------REGISTERED STUDENTS------------');
  for (var s in student){
          s.stdlist();
        }
  print('-----------------------------------------');

```



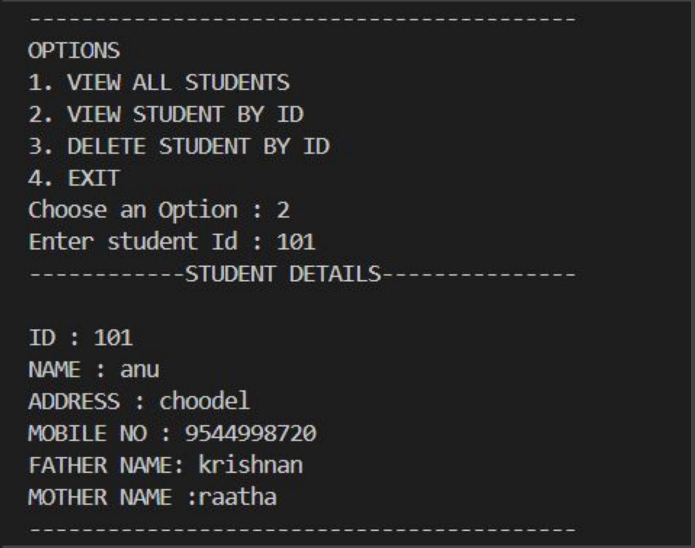
```
-----------------------------------------
OPTIONS
1. VIEW ALL STUDENTS
2. VIEW STUDENT BY ID
3. DELETE STUDENT BY ID
4. EXIT
Choose an Option : 1
----------ALL STUDENTS DETAILS------------

ID : 100
NAME : ram
ADDRESS : kunnamkulam
MOBILE NO : 8976545682
FATHER NAME: raaman
MOTHER NAME :seetha
-----------------------------------------

ID : 101
NAME : anu
ADDRESS : choodel
MOBILE NO : 9544998720
FATHER NAME: krishnan
MOTHER NAME :raatha
-----------------------------------------
```



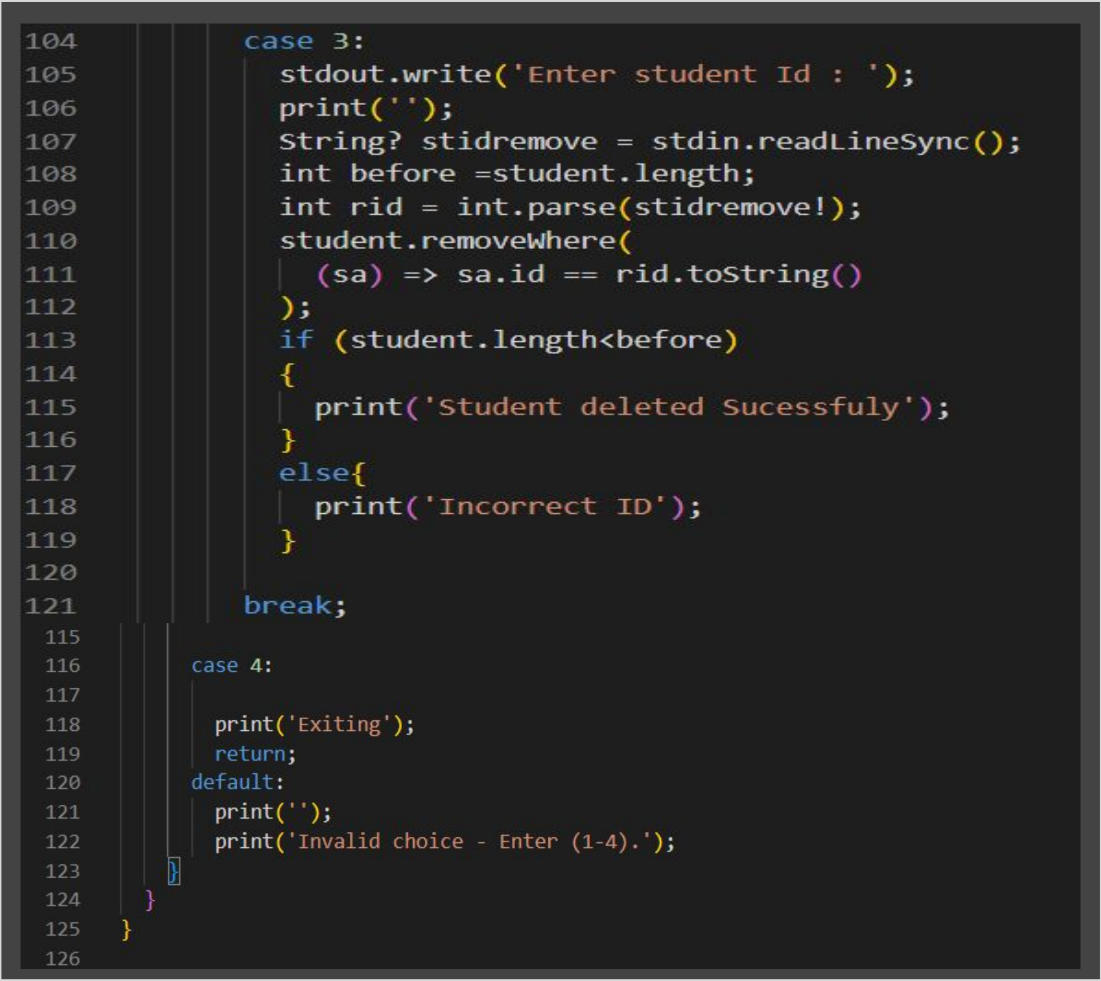
```
while (true){
  print('OPTIONS');
  print('1. VIEW ALL STUDENTS');
  print('2. VIEW STUDENT BY ID');
  print('3. DELETE STUDENT BY ID');
  print('4. EXIT');
  stdout.write('Choose an Option : ');
  String? op = stdin.readLineSync();
  int option = int.parse(op!);
  switch(option){

    case 1:
      print('----------ALL STUDENTS DETAILS------------');
        if(student.isEmpty){
          print("no students found");
        }
        else{
            for (var s in student){
          s.stdlist();
        }
        }
      print('-----------------------------------------');
    break;
```



```
case 2:
  stdout.write('Enter student Id : ');
  String? stidview = stdin.readLineSync();

  if(student.isEmpty){
    print('No students available!');
  }
  else{
    try{
    Student vstd = student.firstWhere(
      (s) => s.id == stidview,
     );
  print('------------STUDENT DETAILS---------------');
    vstd.stdlist();
  print('------------------------------------------');
  }catch(e){
    print('Incorrect ID');
  }
  }

break;
```

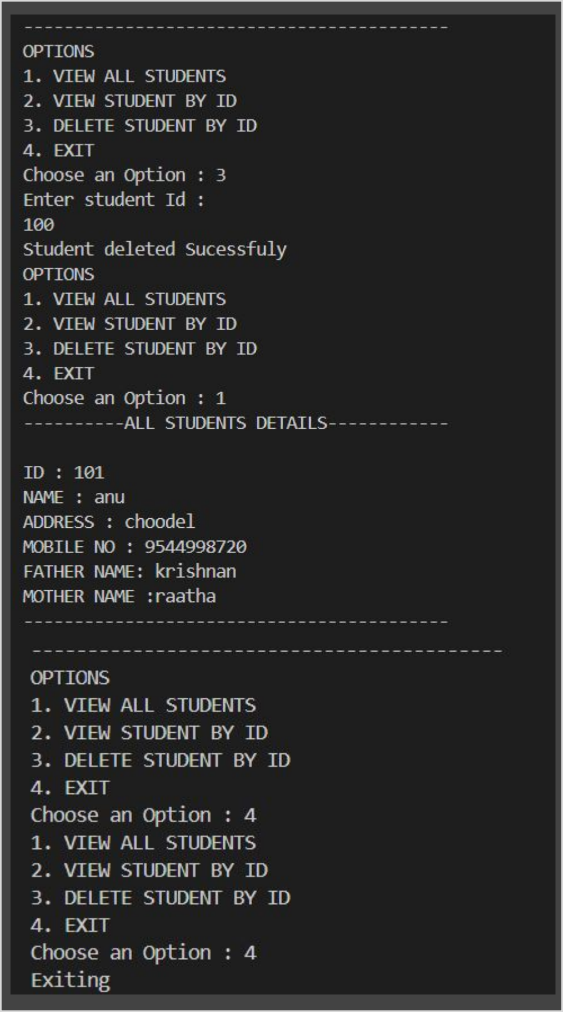


```
------------------------------------------
OPTIONS
1. VIEW ALL STUDENTS
2. VIEW STUDENT BY ID
3. DELETE STUDENT BY ID
4. EXIT
Choose an Option : 2
Enter student Id : 101
------------STUDENT DETAILS---------------

ID : 101
NAME : anu
ADDRESS : choodel
MOBILE NO : 9544998720
FATHER NAME: krishnan
MOTHER NAME :raatha
------------------------------------------
```



```
      case 3:
        stdout.write('Enter student Id : ');
        print('');
        String? stidremove = stdin.readLineSync();
        int before =student.length;
        int rid = int.parse(stidremove!);
        student.removeWhere(
          (sa) => sa.id == rid.toString()
        );
        if (student.length<before)
        {
          print('Student deleted Sucessfuly');
        }
        else{
          print('Incorrect ID');
        }

      break;
```

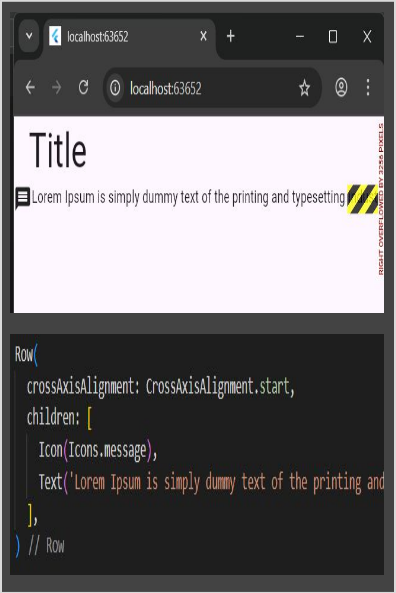
```

      case 4:

        print('Exiting');
        return;
      default:
        print('');
        print('Invalid choice - Enter (1-4).');
      }
    }
  }

```



```
------------------------------------------
OPTIONS
1. VIEW ALL STUDENTS
2. VIEW STUDENT BY ID
3. DELETE STUDENT BY ID
4. EXIT
Choose an Option : 3
Enter student Id :
100
Student deleted Sucessfuly
OPTIONS
1. VIEW ALL STUDENTS
2. VIEW STUDENT BY ID
3. DELETE STUDENT BY ID
4. EXIT
Choose an Option : 1
----------ALL STUDENTS DETAILS-----------

ID : 101
NAME : anu
ADDRESS : choodel
MOBILE NO : 9544998720
FATHER NAME: krishnan
MOTHER NAME :raatha
------------------------------------------
```

```
 ------------------------------------------
 OPTIONS
 1. VIEW ALL STUDENTS
 2. VIEW STUDENT BY ID
 3. DELETE STUDENT BY ID
 4. EXIT
 Choose an Option : 4
 1. VIEW ALL STUDENTS
 2. VIEW STUDENT BY ID
 3. DELETE STUDENT BY ID
 4. EXIT
 Choose an Option : 4
 Exiting
```

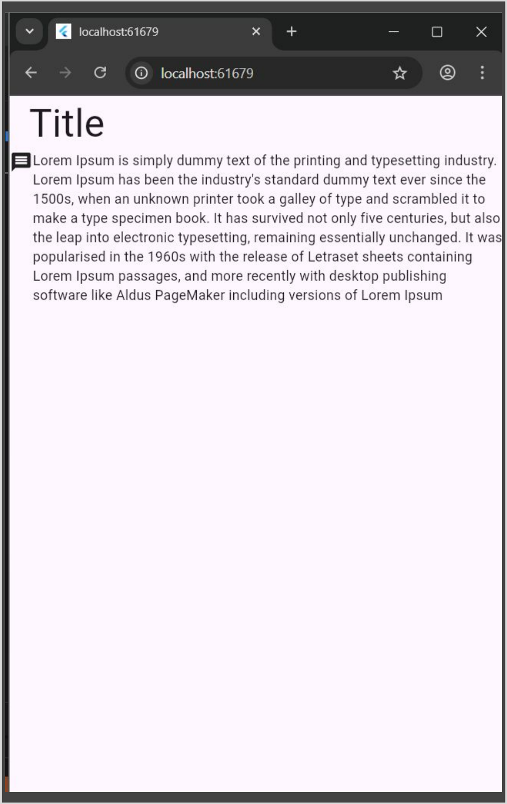
Exercise 2



Title

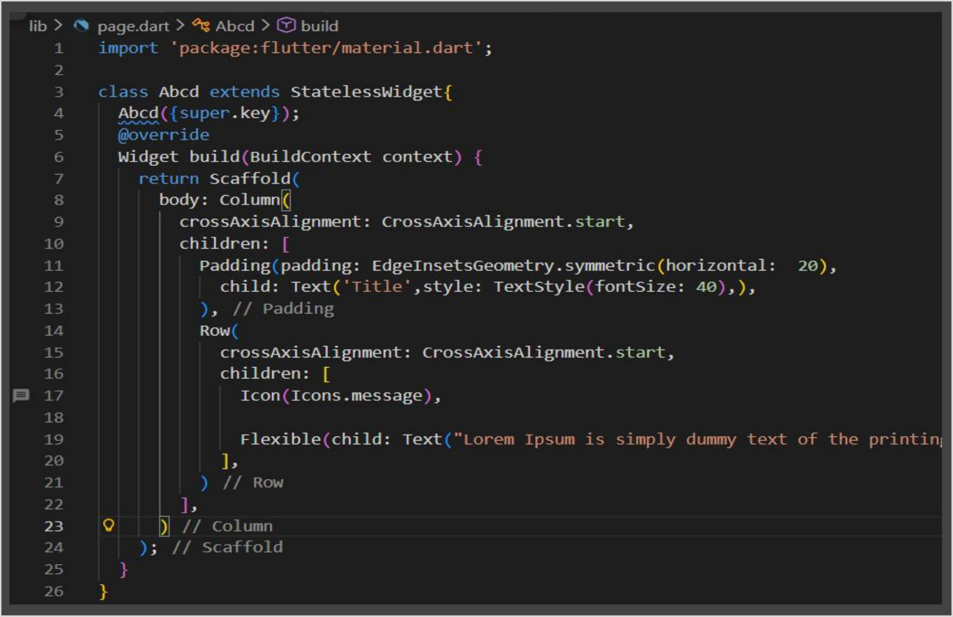
Lorem Ipsum is simply dummy text of the printing and typesetting

```
Row(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Icon(Icons.message),
    Text('Lorem Ipsum is simply dummy text of the printing and
  ],
) // Row
```



Title

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum



```
import 'package:flutter/material.dart';

class Abcd extends StatelessWidget{
  Abcd({super.key});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Padding(padding: EdgeInsetsGeometry.symmetric(horizontal:  20),
            child: Text('Title',style: TextStyle(fontSize: 40),),
          ), // Padding
          Row(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Icon(Icons.message),

              Flexible(child: Text("Lorem Ipsum is simply dummy text of the printin
            ],
          ) // Row
        ],
      ) // Column
    ); // Scaffold
  }
}
```

FLEXIBLE

THANK YOU !