UNIVERSITY OF RWANDA

COLLEGE OF SCIENCE AND TECHNOLOGY

GAKO CAMPUS

COMPUTER ENGINEERING

MOBILE APPLICATIONS

DATE ON 15th feb,2026

Group members :-Abdoul SIBO 223026681

-JUSTINE IRAFASHA 223026693

-FILS NIYINZIMA 223026876

DART LAB REPORT (lab 1)

Part 1: Functions

Q1: Write a Dart function named `welcomeMessage` that prints a welcome message for the school system.

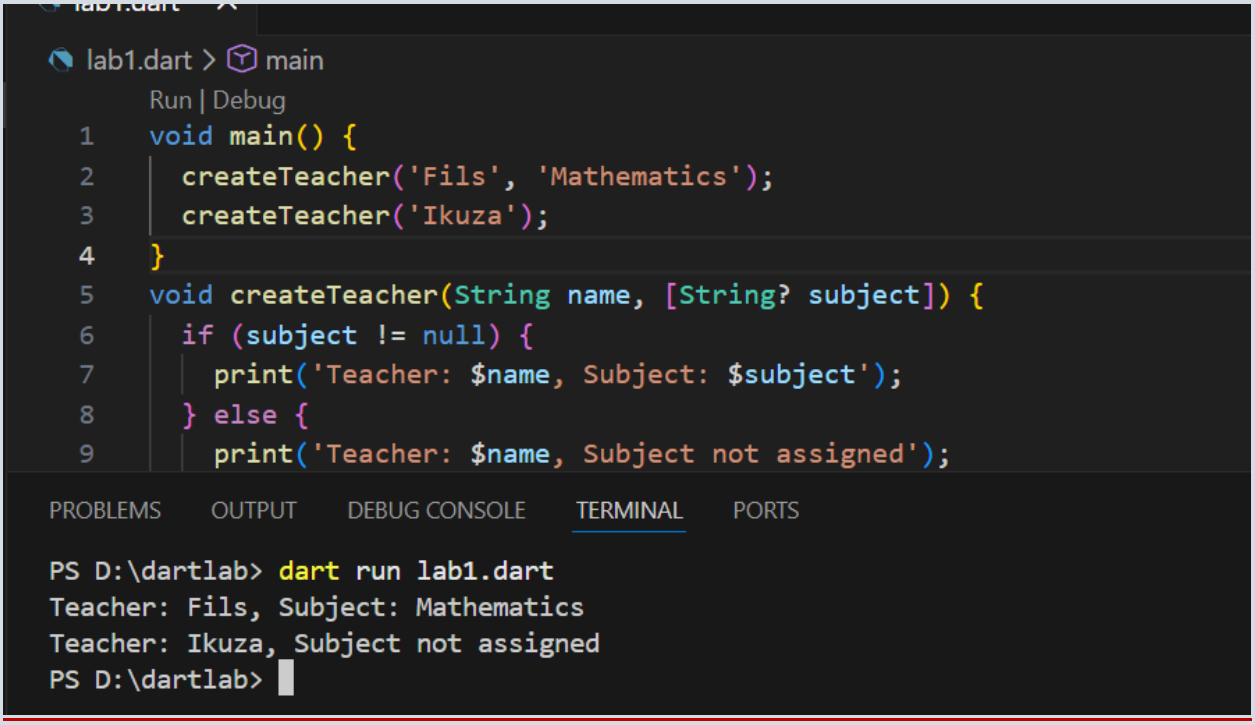
Answer screenshot 1:

Q2: Write a function named `createStudent` that uses named parameters (`name` and `age`) and prints the student details.

Answer screenshot 2:

Q3: Write a function named `createTeacher` with one required parameter `name` and one optional parameter `subject`. If subject is not provided, print 'Subject not assigned'.

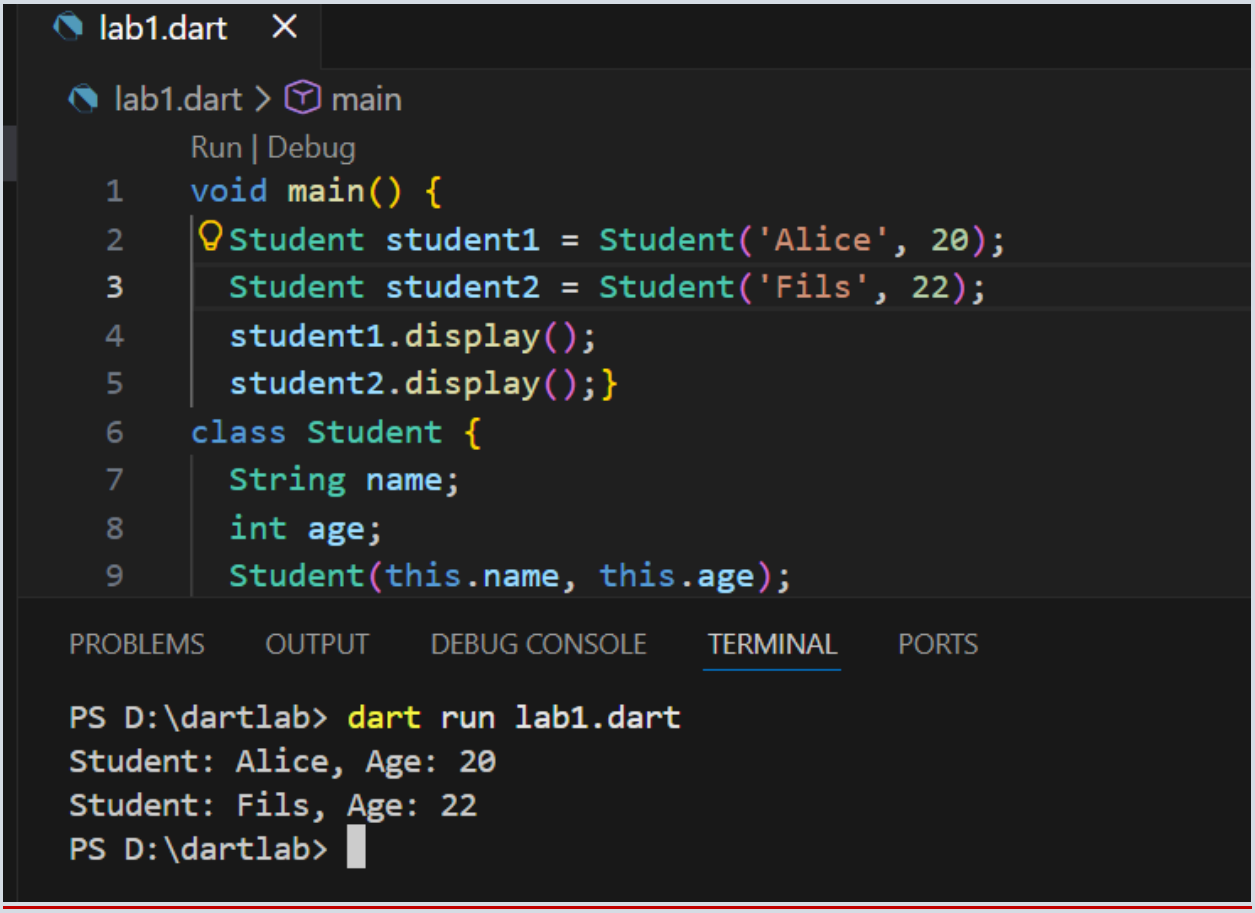
Answer screenshot 3:



Part 2: Constructors and Classes

Q4: Create a class named `Student` with `name` and `age`, and a constructor to initialize these values. Together with Q5: Create an object of `Student` and print

the student’s name and age **.Screenshot**

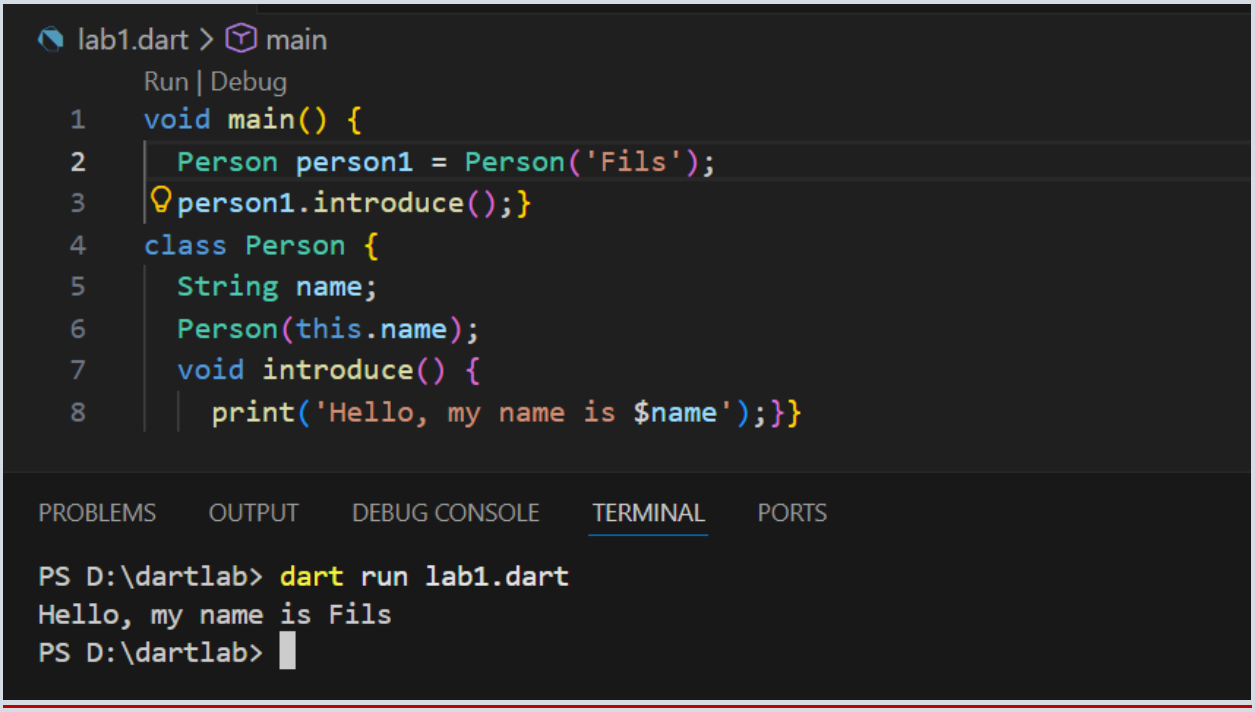


Part 3: Inheritance

Q6: Create a class `Person` with a variable `name` and a function `introduce()` that prints

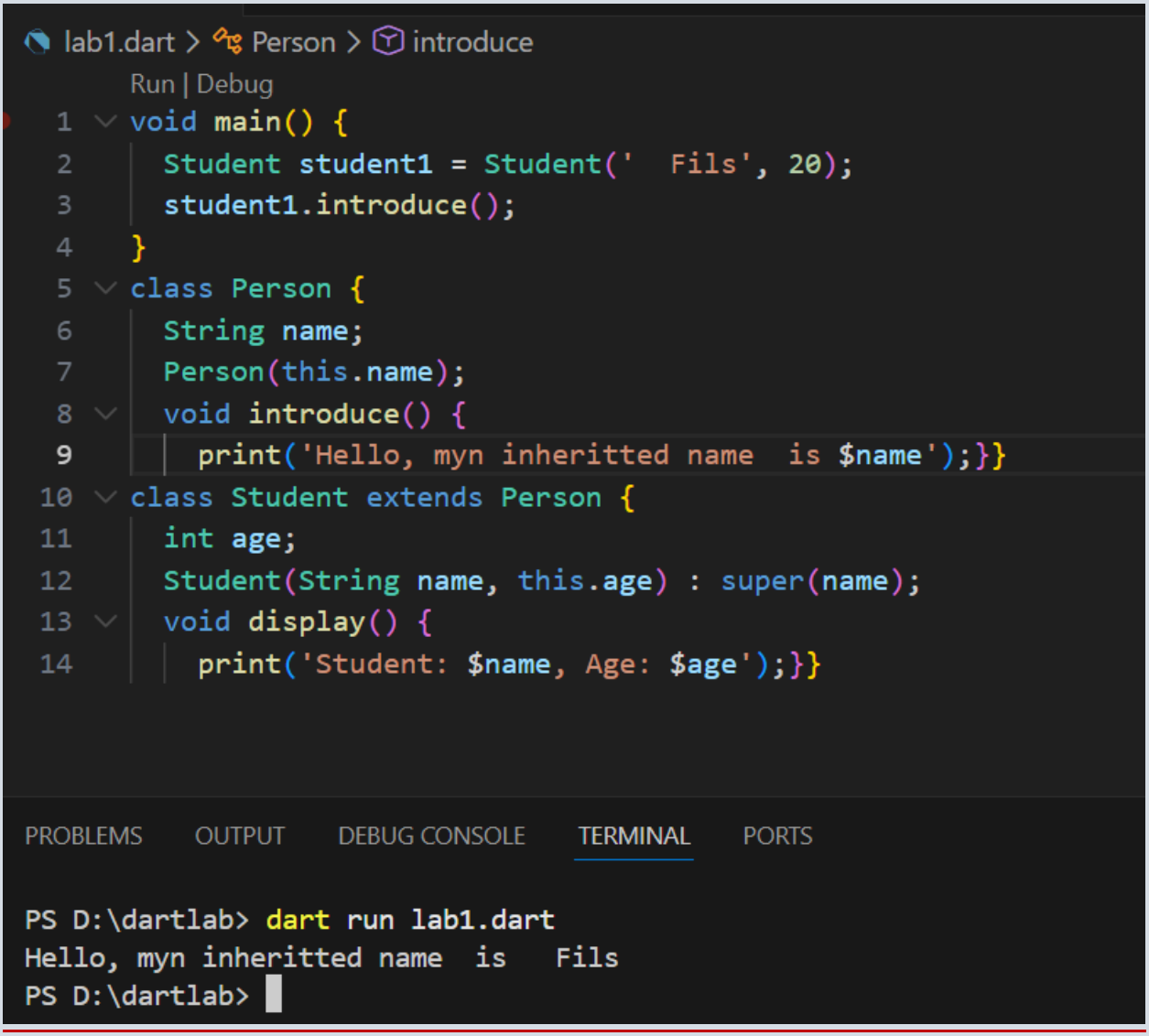
the name.

screenshot for Answer 6:



Q7: Make `Student` inherit from `Person` and call `introduce()` from a `Student` object.

Answer screenshot 7:



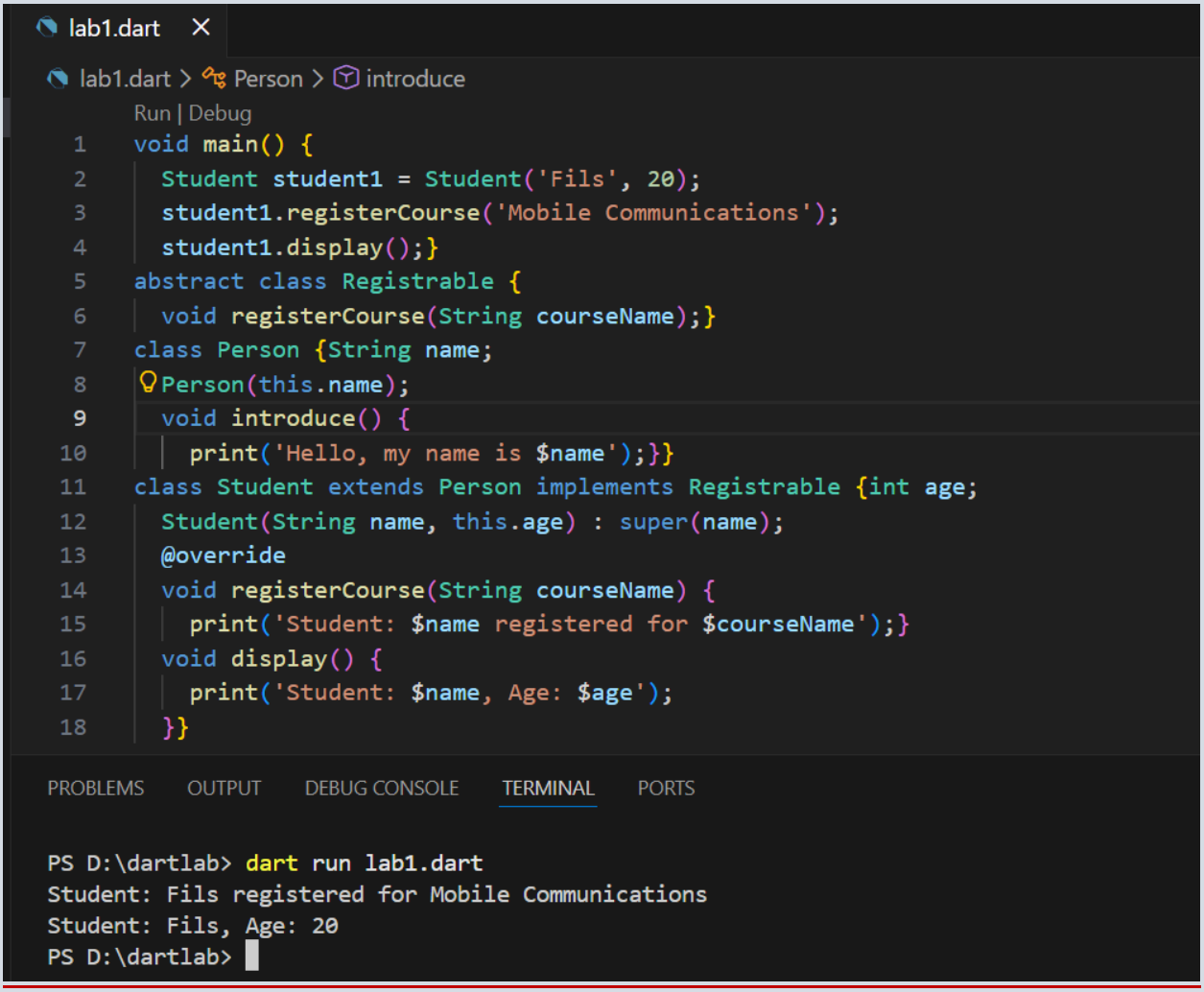
Part 4: Interfaces

Q8: Create an abstract class `Registrable` with a function `registerCourse(String courseName)`. **Together combined with**

Q9: Make `Student` implement `Registrable` and implement `registerCourse` to print the

student name and course.

Answer screenshot for q8 & q9:

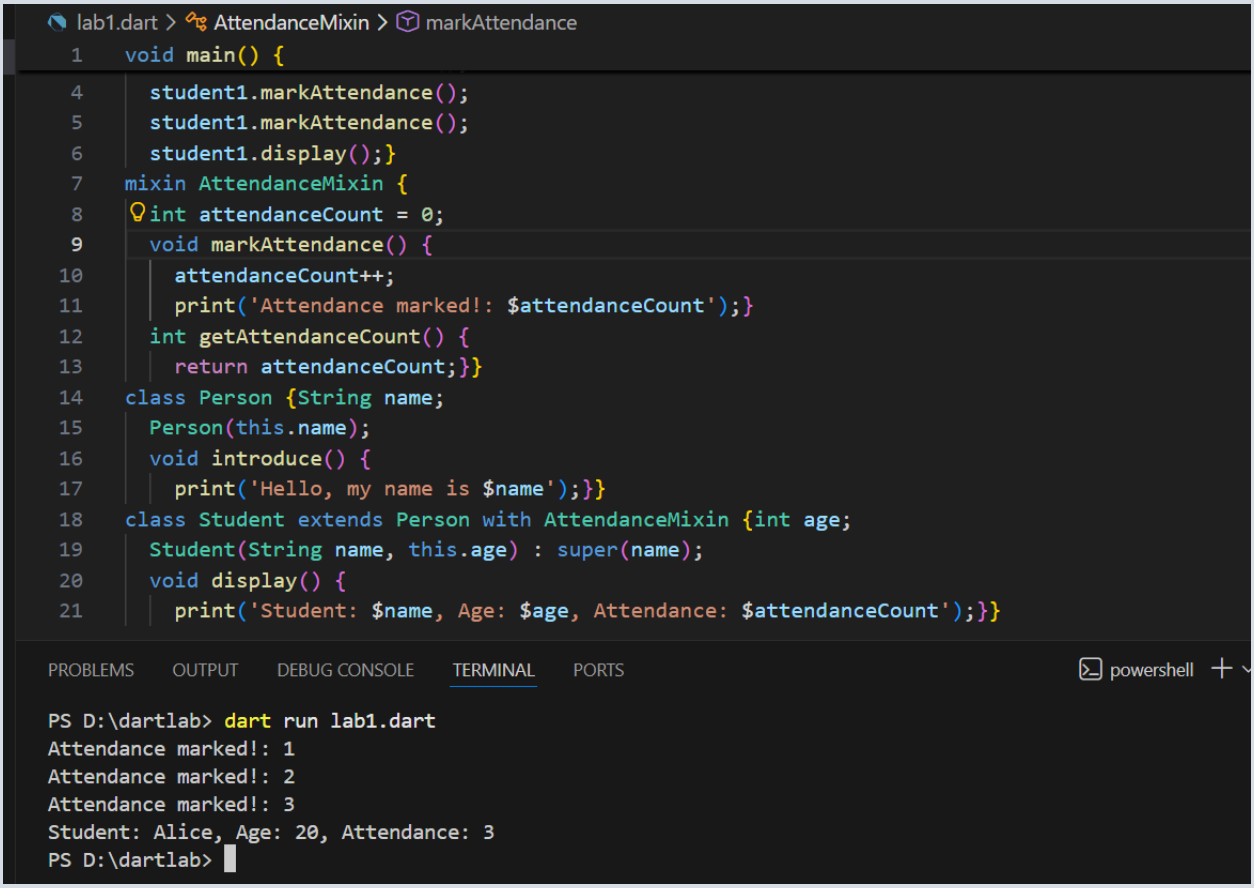


Part 5: Mixins

Q10: Create a mixin `AttendanceMixin` that stores an attendance counter and has a function `markAttendance()` to increase attendance.

Q11: Apply `AttendanceMixin` to `Student`. Mark attendance 3 times and print the attendance.

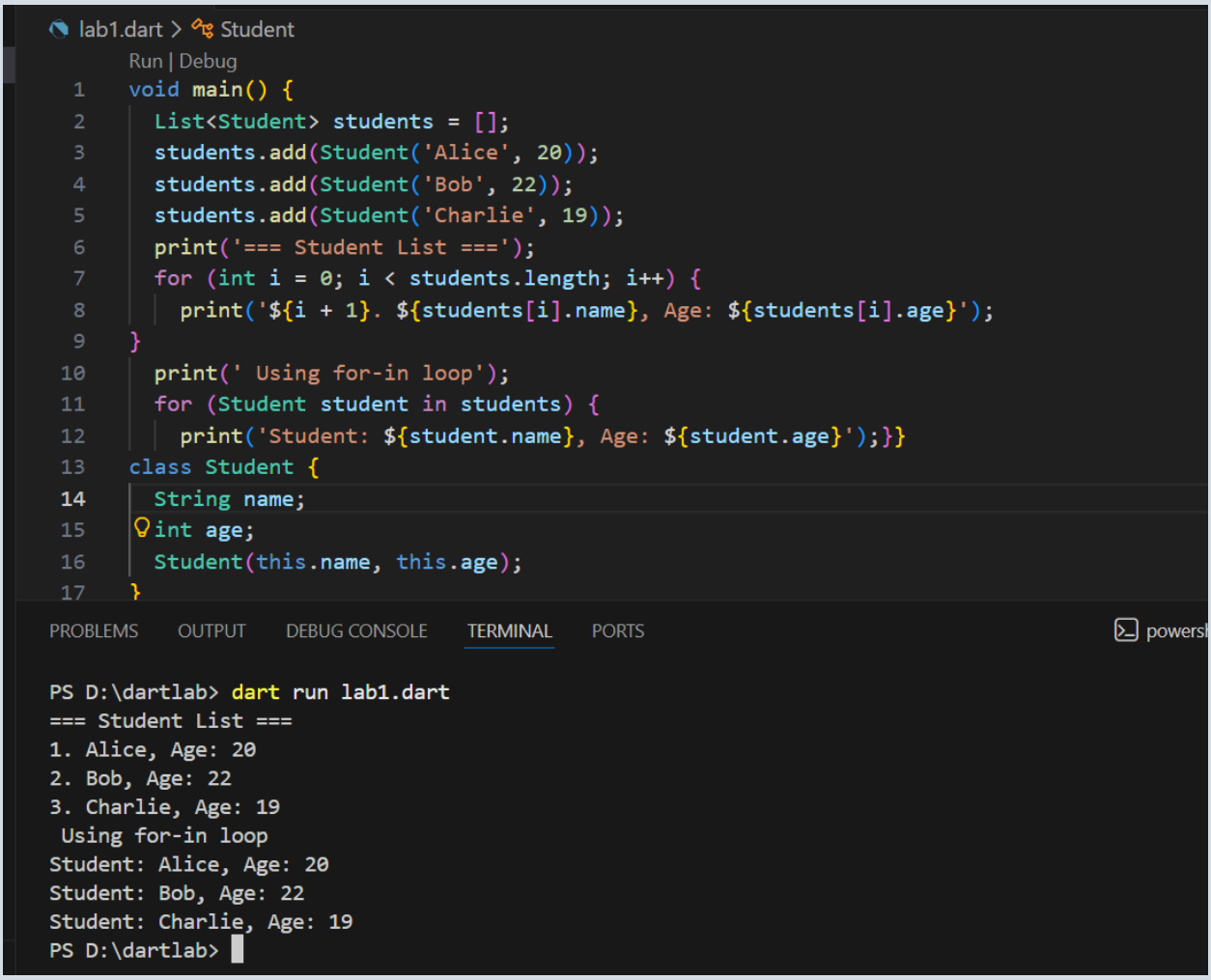
Answer screenshot q10 & q11:



Part 6: Collections

Q12: Create a List storing multiple `Student` objects. Add 3 students.

Answer screenshot 12:



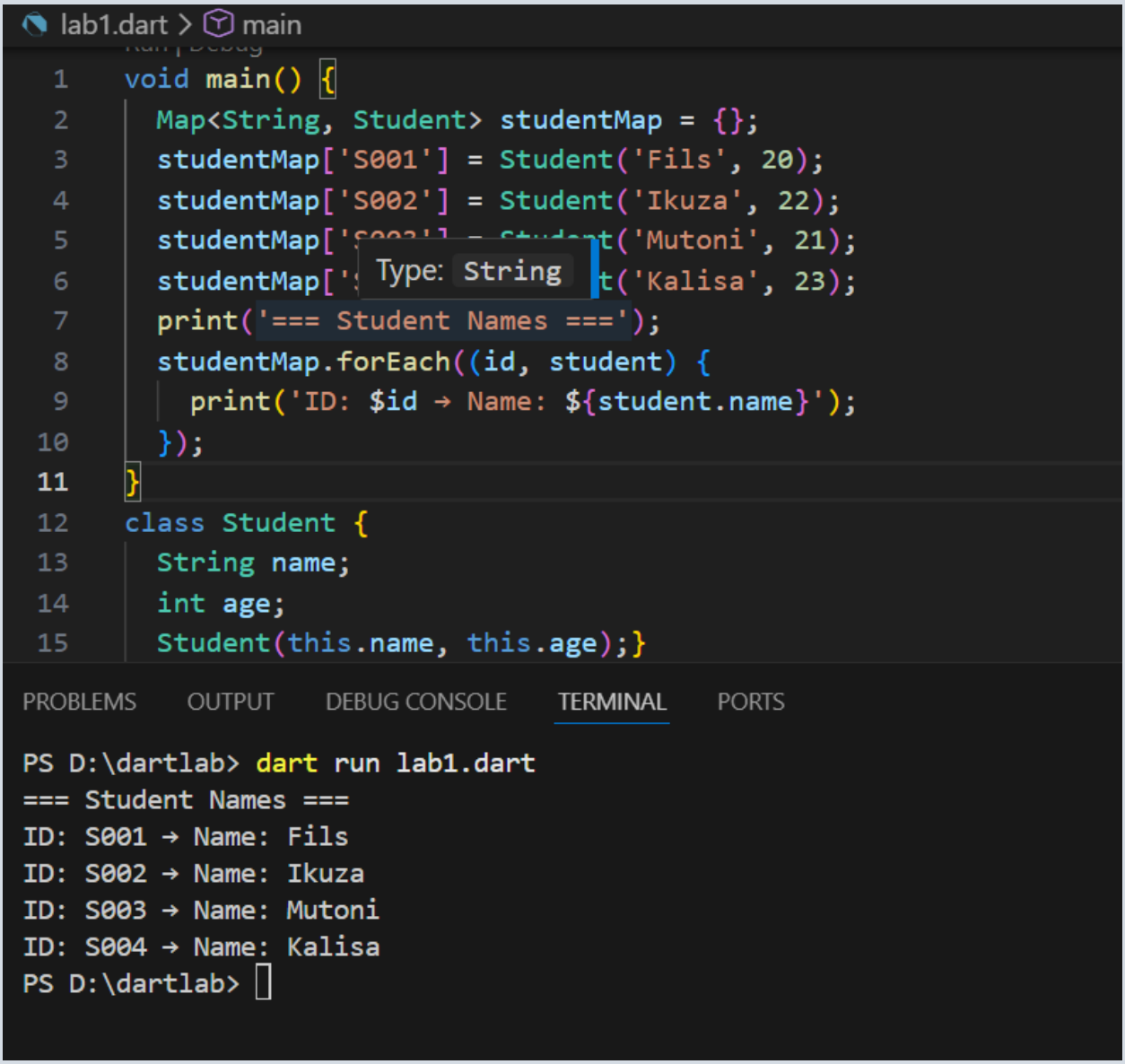
```
lab1.dart > Student
    Run | Debug
1   void main() {
2     List<Student> students = [];
3     students.add(Student('Alice', 20));
4     students.add(Student('Bob', 22));
5     students.add(Student('Charlie', 19));
6     print('=== Student List ===');
7     for (int i = 0; i < students.length; i++) {
8       print('${i + 1}. ${students[i].name}, Age: ${students[i].age}');
9   }
10    print(' Using for-in loop');
11    for (Student student in students) {
12      print('Student: ${student.name}, Age: ${student.age}');}}
13  class Student {
14    String name;
15    int age;
16    Student(this.name, this.age);
17  }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powersh

PS D:\dartlab> dart run lab1.dart
=== Student List ===
1. Alice, Age: 20
2. Bob, Age: 22
3. Charlie, Age: 19
 Using for-in loop
Student: Alice, Age: 20
Student: Bob, Age: 22
Student: Charlie, Age: 19
PS D:\dartlab>
```

Q13: Create a Map where the key is student ID and value is a `Student`. Print all student names.

Answer screenshot 13:



Part 7: Anonymous and Arrow Functions

Q14: Use an anonymous function to print all student names from the list.

Answer screenshot 14:

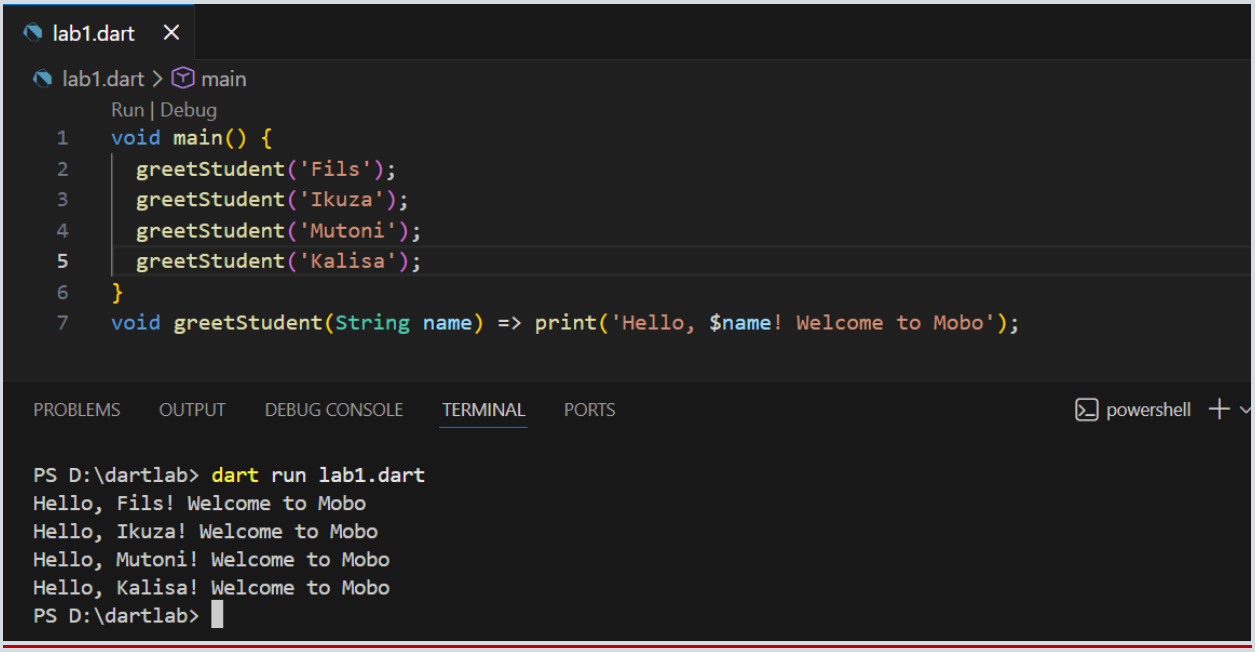
```
lab1.dart > main
    Run | Debug
1   void main() {
2     List<Student> students = [
3       Student('Fils', 20),
4       Student('Ikuza', 22),
5       Student('Mutoni', 21),
6       Student('Kalisa', 23),];
7     print('Student Names Anonymouslyare:');
8     students.forEach((Student student) {
9       print(student.name);
10    });}
11  class Student {
12    String name;
13    int age;
14
15    Student(this.name, this.age);
16  }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\dartlab> dart run lab1.dart
Student Names Anonymouslyare:
Fils
Ikuza
Mutoni
Kalisa
PS D:\dartlab>
```

Q15: Write an arrow function that takes a student name and prints a greeting message.

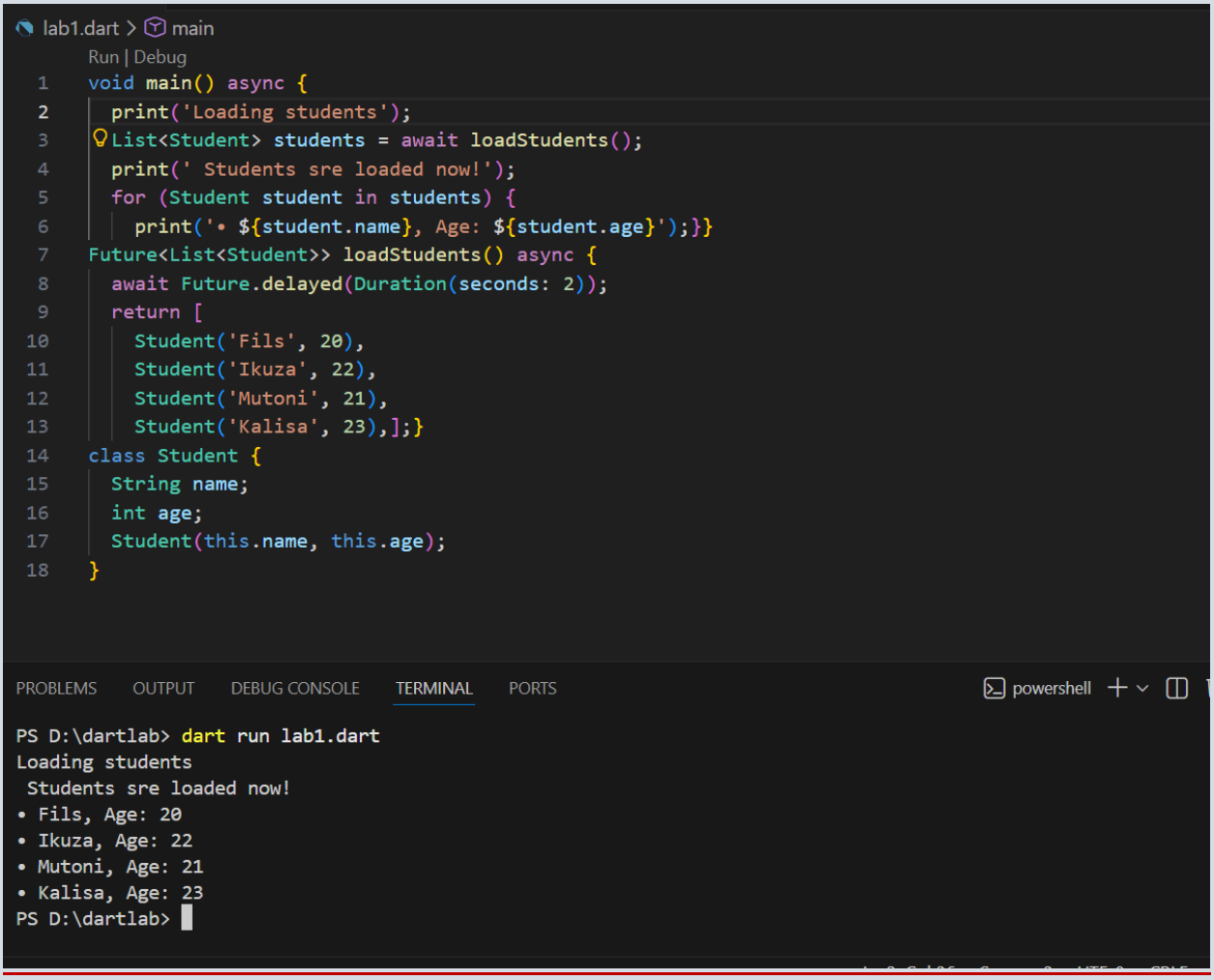
Answer screenshot15 :



Part 8: Asynchronous Programming

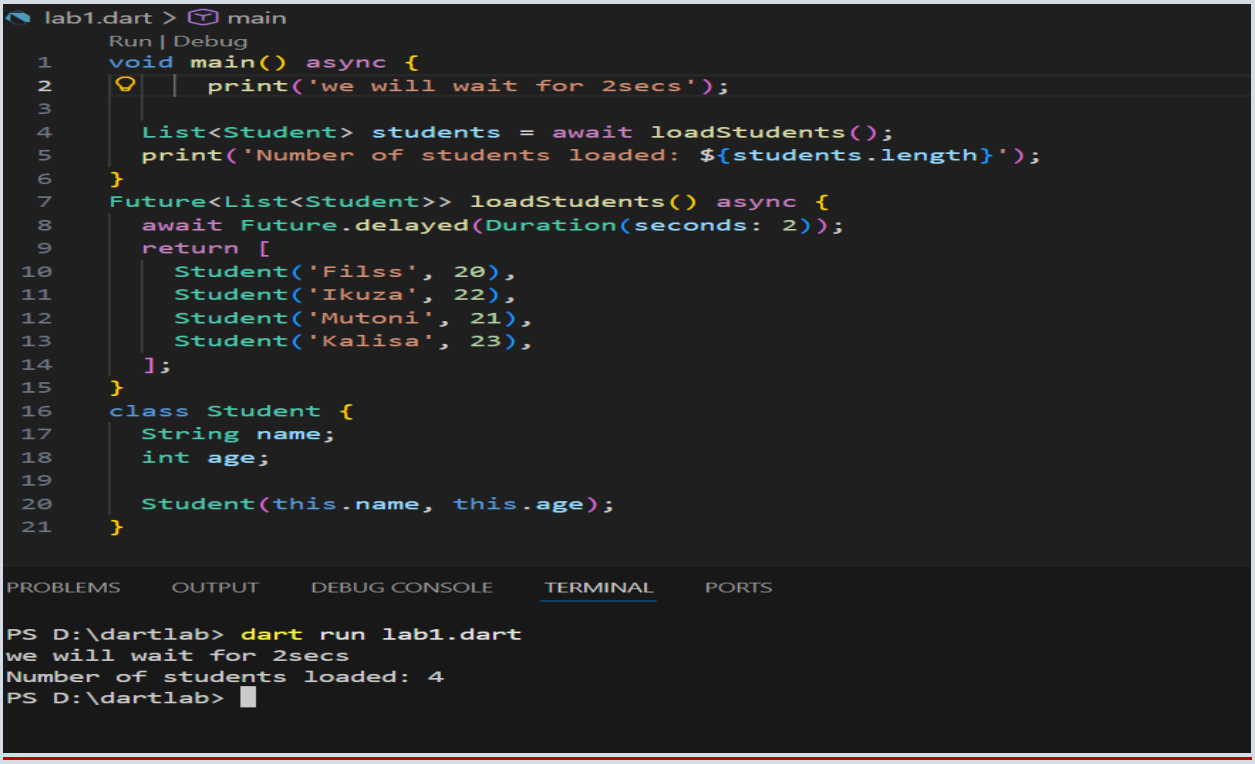
Q16: Write an async function `loadStudents()` that waits 2 seconds and returns the list of students.

Answer screenshot 16:



Q17: In main(), call `loadStudents()`, use await, and print the number of students loaded.

Answer screenshot 17 :



Part 9: Integration Challenge

Q19: Create a new mixin `NotificationMixin` that prints a message when a student is registered to a course. Apply it to `Student`.

Answer screenshot 19 :

lab1.dart > Person

```dart
void main() {
  Student student1 = Student('Fils', 20);
  student1.registerCourse('Mobo');
  student1.registerCourse('NetSec');
  print(' Student Summary ');
  student1.display();}
mixin NotificationMixin {
  void sendNotification(String studentName, String courseName) {
    print(' NOTIFICATION: $studentName has successfully registered for $courseName!');
    print('  Registration confirmed. Welcome to the course!');}}
abstract class Registrable {
void registerCourse(String courseName);}
class Person {String name;
  Person(this.name);
  void introduce() {
    print('Hello, my name is $name');}}
class Student extends Person with NotificationMixin implements Registrable { int age;
  Student(String name, this.age) : super(name);
  @override
  void registerCourse(String courseName) {
    sendNotification(name, courseName);}
  void display() {
    print('Student: $name, Age: $age');}}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

```
PS D:\dartlab> dart run lab1.dart
 NOTIFICATION: Fils has successfully registered for Mobo!
  Registration confirmed. Welcome to the course!
 NOTIFICATION: Fils has successfully registered for NetSec!
  Registration confirmed. Welcome to the course!
 Student Summary
Student: Fils, Age: 20
```