// Exercise 0: Simple Interest

void calculateSimpleInterest(double p, double t, double r) {

double simpleInterest = (p \* t \* r) / 100;

print('Simple Interest: \$${simpleInterest.toStringAsFixed(2)}');

}

// Exercise 1: Simple Calculator

class Calculator {

double add(double a, double b) => a + b;

double subtract(double a, double b) => a - b;

double multiply(double a, double b) => a \* b;

double divide(double a, double b) => a / b;

}

// Exercise 2: Print 1 to 100 but not 41

void printNumbersExcept(int num) {

for (int i = 1; i <= num; i++) {

if (i != 41) {

print(i);

}

}

}

// Exercise 3: createUser function

void createUser(String name, int age, {bool isActive = true}) {

print('Name: $name, Age: $age, Active: $isActive');

}

// Exercise 4: Find a name that starts with 'a'

void findNameWithA(List<String> friends) {

String? friend = friends.firstWhere((friend) => friend.startsWith('a'), orElse: () => '');

print('Friend name that starts with A: $friend');

}

// Exercise 5: Laptop class

class Laptop {

int id;

String name;

int ram;

Laptop(this.id, this.name, this.ram);

@override

String toString() => 'ID: $id, Name: $name, RAM: $ram GB';

}

void main() {

// Exercise 5: Create 3 Laptop objects

List<Laptop> laptops = [

Laptop(1, 'Dell', 8),

Laptop(2, 'HP', 16),

Laptop(3, 'Lenovo', 12),

];

laptops.forEach((laptop) => print(laptop));

}

import 'dart:async';

import 'dart:io';

import 'package:path/path.dart' as path;

// Exercise 6: Interface and Class

abstract class Bottle {

void open();

}

class CokeBottle implements Bottle {

@override

void open() {

print('Coke bottle is opened');

}

static Bottle create() {

return CokeBottle();

}

}

void main() {

// Exercise 6: Instantiate CokeBottle using the factory constructor and call open()

Bottle bottle = CokeBottle.create();

bottle.open();

}

// Exercise 7: Late variable

late String address = 'US';

void main() {

// Exercise 7: Print late variable

print(address);

}

// Exercise 8: Function with nullable int parameter

int returnZeroIfNull(int? value) {

return value ?? 0;

}

void main() {

// Exercise 8: Print result of function with nullable int parameter

int? nullableValue = null;

print(returnZeroIfNull(nullableValue));

}

// Exercise 9: Read CSV file and print its content

void readAndPrintCSVFile(String filePath) async {

final File file = File(filePath);

List<String> lines = await file.readAsLines();

lines.forEach((line) => print(line));

}

// Exercise 10: Sort list asynchronously

void sortList(List<String> items) async {

List<String> sortedList = await Future.delayed(Duration(seconds: 1), () {

items.sort();

return items;

});

print(sortedList);

}

void main() {

// Exercise 10: Sort list asynchronously

List<String> items = ['banana', 'apple', 'orange'];

sortList(items);

}

// Exercise 11: Simple Quiz Application

class Question {

String questionText;

bool answer;

Question(this.questionText, this.answer);

}

class Quiz {

List<Question> questions;

int \_currentQuestionIndex = 0;

int score = 0;

Quiz(this.questions);

void nextQuestion() {

if (\_currentQuestionIndex < questions.length - 1) {

\_currentQuestionIndex++;

} else {

print('Quiz completed! Your score is: $score');

}

}

void answerQuestion(bool userAnswer) {

if (userAnswer == questions[\_currentQuestionIndex].answer) {

score++;

}

nextQuestion();

}

void start() {

print('--- Quiz Start ---');

for (var question in questions) {

print(question.questionText);

String userInput = stdin.readLineSync() ?? '';

bool userAnswer = userInput.toLowerCase() == 'true';

answerQuestion(userAnswer);

}

}

}

void main() {

// Exercise 11: Create a simple quiz application

List<Question> questions = [

Question('Dart is a programming language.', true),

Question('Flutter is used for web development.', false),

Question('Java is a statically typed language.', true),

];

Quiz quiz = Quiz(questions);

quiz.start();

}