

**SEB-312 Mobile Application Development**

**LAB # 02**

**LAB Title**

|  |
| --- |
| Writing simple Dart programs: Variables, loops, and functions, implementing basic OOP concepts: Creating classes, objects, and methods., Manipulating Lists, Sets, and Maps, implementing asynchronous functions with Future and async/await, |

**Assessment of CLO: 03, PLO: 05**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** |  | | |
| **Roll No.** |  | | |
| **Semester** |  | **Session** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Perf. Level**  **Criteria** | **Excellent**  **(2.5)** | **Good**  **(2)** | **Satisfactory**  **(1.5)** | **Needs Improvement**  **(0 ~ 1)** | **Marks Obtained** |
| **1** | Project Execution & Implementation | Fully functional, optimized, and well-structured. | Minor errors, mostly functional. | Some errors, requires guidance. | Major errors, non-functional, or not Performed. |  |
| **2** | Results & Debugging  Or Troubleshooting | Accurate results with effective debugging  Or Troubleshooting. | Mostly correct, some debugging Or Troubleshooting needed. | Partial results, minimal debugging  Or Troubleshooting. | Incorrect results, no debugging Or Troubleshooting, or not attempted. |  |
| **3** | Problem-Solving & Adaptability  (VIVA) | Creative approach, efficiently solves challenges. | Adapts well, minor struggles. | Some adaptability, needs guidance. | Lacks innovation or no innovation, unable to solve problems. |  |
| **4** | Report Quality & Documentation | Clear, structured, with detailed visuals. | Mostly clear, minor gaps. | Some clarity issues, missing details. | Poorly structured, lacks clarity, or not submitted. |  |
| **Total Marks Obtained Out of 10** | | | | | |  |

**Experiment evaluated by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor’s Name** | **Sidra Khatoon** | | |
| **Date** |  | **Signature** |  |

**Objective**

The objective of this lab session is to practice Dart language

**Assessments**

1. Create a constant named myAge and initialize it with your age. Write an if statement to print out “Teenager” if your age is between 13 and 19 , and “Not a teenager” if your age is not between 13 and 19.

Ans:

void main() {

  const int myAge = 20;

  if (myAge >= 13 && myAge <= 19) {

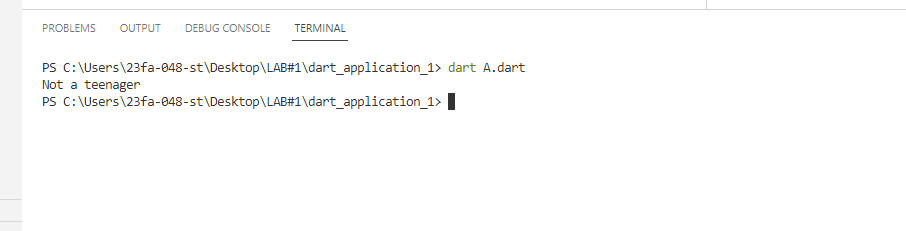
    print("Teenager");

  } else {

    print("Not a teenager");

  }

}



1. Write a function named youAreWonderful , with a string parameter called name . It should return a string using name , and say something like “You’re wonderful, Bob.”

Ans:

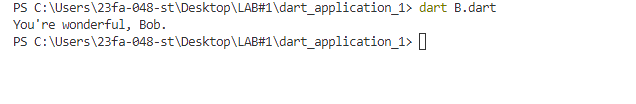
String youAreWonderful(String name) {

  return "You're wonderful, $name.";

}

void main() {

  print(youAreWonderful("Bob")); }



1. Add another int parameter to above function called numberPeople so that the function returns something like “You’re wonderful, Bob. 10 people think so.”

Ans:

String youAreWonderful(String name, int numberPeople) {

  return "You're wonderful, $name. $numberPeople people think so.";

}

void main() {

  print(youAreWonderful("Bob", 10)); // Example usage

}



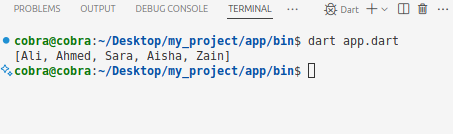
4.Create a list of names and print all names using list.

void **main**() {

**List**<**String**> names = ['Ali', 'Ahmed', 'Sara', 'Aisha', 'Zain'];

**print**(names);

}



5.Create a program that’s reads list of expenses amount using user input and print total.

import 'dart:io';

void main() {

List<double> expenses = [];

print('Enter number of expenses:');

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

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

print('Enter expense ${i + 1}:');

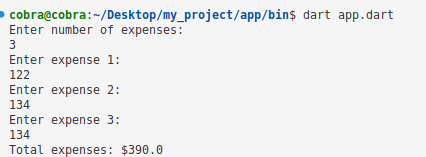
double expense = double.parse(stdin.readLineSync()!);

expenses.add(expense); }

double total = expenses.reduce((sum, element) => sum + element);

print('Total expenses: \$${total}');

}



6.Create an empty list of type string called days. Use the add method to add names of 7 days and print all days.

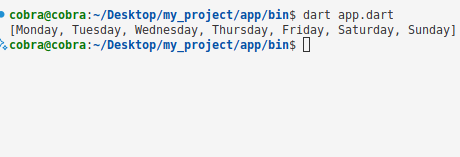
void main() {

List<String> days = [];

days.addAll(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']);

print(days);

}



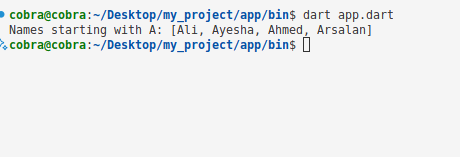
7.Add your 7 friend names to the list. Use where to find a name that starts with alphabet a.

void main() {

List<String> friends = ['Ali', 'Bilal', 'Ayesha', 'Sara', 'Ahmed', 'Zain', 'Arsalan'];

var aNames = friends.where((name) => name.startsWith('A')).toList();

print('Names starting with A: $aNames');}



8.Create a map with name, address, age, country keys and store values to it. Update country name to other country and print all keys and values.

void main() {

Map<String, dynamic> person = {

'name': 'Abdullah',

'address': 'Street 123, Karachi',

'age': 22,

'country': 'Pakistan'

};

person['country'] = 'Germany'; // Updating country name

person.forEach((key, value) {

print('$key: $value');

});

}

