GROUP 15 5TH FEB 2026

Names: Aline NIYONIZERA
Gaston UWIZEYIMANA

Course: Mobile Application Systems And Design.

Unit Programming Lab 1 Report

Q1. A function is a block of code that performs a specific task
Creating a function helps organise the program and allows the message to be reused easily.
This program defines a function called WelcomeMessage
The key concept used is a function.
We learned how to create and call a function in Unity to organise program output.

Q2.

```
Void WelcomeMessage () {
    Print ("======");
    Print ("WELCOME TO THE SCHOOL SYSTEM");
    Print ("======");
    Print ("This system helps manage School records");
    Print ("======");
}
```

Q2 This function Print Student details using named Parameters.

```
Void Greatestudent ( required String name, required int age ){
    {
    Print ("Student details");
    Print ("Name: $name");
    Print ("Age: $age");
    }.
```

=> Named Parameters are Parameters that are passed to a function by name instead of position they are written inside curly brackets {} in a function definition.
They make function calls clear and readable.

→ We learned how to pass data to functions using named Parameters.

Q3 This function has one required Parameter and one Optional Parameter.
Optional Parameters are Parameters that may or may not be provided when calling a function.
If Optional Parameter is not given the function can use a default value or handle it safely.
From this Question We learned how dart handle optional values without causing Errors.

Q4 A Constructor is a special function used to initialise a Class object.
In this Code, the Constructor assigns Value to the name and age of student
Constructor is important because they ensure object start
with correct data.
From this Question we learned how to initialise class variables using
Constructor in Dart

Q5 An object is an instance of a Class created using a Constructor
In this Question a student object is created and used to access its data
Object Creation allows a Program to work with real data values.
We learned how to create and use objects data.

Q6 A Class is a blue Print used to Create object
It groups Variables and functions that belong together
In Question the Person Class store a name and print it
We learned how Classes organise data and behaviour in Dart.

Q7. Inheritance allows One Class to reuse another Class's properties and
Methods.
The Student Class inherits from Person class
This allows Student to use the introduce() method without rewriting
it
We learned how Inheritance reduce Code duplication.

Q8 An Interface defines methods that a Class must implement.
In short an abstract Class act as an Interface
It does not provide implementation, only method definitions
We learned how Interfaces define rules for classes

Q9 Implementing an Interface means providing all required method
The Student Class implements Registrable Interface.
This enforces the rule that registerCourse() must be defined.
We learned how Interfaces ensure Consistency in classes.

Q10. A mixin is a way to add functionality to a Class without using Inheritance
It allows sharing methods and variables across multiple classes.
In this Question mixin adds attendance tracking behavior.
We learned how mixins help reuse functionality in bits.

Controlled in dart

Q11 The mixin adds an Attendance Counter and a function to mark Attendance.

The key concept used is mixins, which allow a class to gain additional behaviour without using inheritance.

We learned how mixins add additional functionality to class.

Q12. Lists are Collections that Allow storing and managing multiple values of same type.

The code Creates a list to store multiple Student objects.

We learned How to Create, store and iterate through multiple objects in a single Collection.

Q13 Map Allows fast access value using unique key.

This code Creates a map where each Student ID is key and value is a Student object.

We learne how maps store key-value data and how to access all values efficiently.

Q14 An anonymous function is a function without name used directly where it is needed.

In this code it prints all student names from a list

We learned how to write short inline function without creating a separate function name.

Q15 An arrow function is a short form of a function that contains a single expression. In this code it prints message for a student.

We learned how arrow function simplify code for single expression operations.

Q16. Async functions allows operations that take time to run without blocking the program.

We learned asynchronous programing, how best handles task waiting using async and await.

Q17 The main function calls async function using await to wait for data.

This ensures the program prints the number of students only after loading

We learned how async programing helps in real apps to handle time consuming tasks efficiently.

Q18. Mixins are useful because they allow classes to reuse behavior from another class without forming a parent-child relationship.

This means a class can gain additional features like methods or properties without being forced into inheritance hierarchy.

Inheritance creates hierarchical relationship where a subclass inherits all the properties and methods of parent class.

The main difference is that inheritance defines "is-a" relationships while mixins add capabilities.

Feature	Inheritance	Mixins.
Relationship	"is-a" relationship (subclass is a)	No relationship, just add features.
Code Reuse	Reuses parent class properties/ method.	Add reusable behavior without hierarchy.
Multiple use	Single inheritance	Can be applied to multiple classes easily.
Purpose	Define class hierarchy	Add functionality to existing classes.
Flexibility.	Less flexible. bound by hierarchy	More flexible. Can mix in only what's needed.
Example	Class Student extend Person	Class Student with AttendanceMixin

Q19 This Code Creates a NotificationMixin to print a message
when a Student registers for a Course
The StudentClass uses mixin while the will respond to
gain notification behavior.

We used mixin inside the registerCourse method so that whenever
Student registers, message is automatically printed!

We Learned How Mixins make it easy to add reusable features
to multiple Classes in Dart.

Q20. Learning Dart helps Understand Flutter because Flutter
apps are written entirely in Dart.
Concepts Such as functions, Classes, objects, Constructors, Collections
async programming and mixins, in Dart are directly used
in flutter development.
Understanding Dart makes it Easier to Create Widgets,
manage State and handle User interactions in flutter.