REG NO: - 22300678 5
- 22302133 7

Dart Programming Lab Report

PART 1: FUNCTIONS

Q1. A Function is a reusable block of code that performs a specific task. The 'WelcomeMessage' function was created to display a greeting for the school system, making the program organised and avoiding repeated code.

Q2. Named parameters allow values to be passed using parameter names, improving readability and reducing errors. They are helpful when a function has multiple parameters and clarity is important.

Q3. Optional parameters allow a function to accept values that may or may not be provided. If an optional value is missing, the function can use a default behavior, making it flexible and adaptable.

PART 2: CONSTRUCTORS and CLASSES

Q4. A Constructor is a special method used to initialize object values when a class object is created.
Constructors ensure that objects start with valid and required data.

Q5. Object creation means creating an instance of a class using the constructor. Objects allows access to class properties and methods. ~~The 'Person' class defined a name variable~~
A 'Student' object was created using the constructor, and its name and age were printed using dot notation.

PART 3: INHERITANCE

Q6. A Class is a blueprint for creating objects. The 'Person' class defined a name variable and an 'introduce()' function to represent a person.

Q7. Inheritance allows a class to use (reuse) properties and methods from another class.
The 'Student' class inherits from 'Person', reducing code duplication and improving reusability.

PART 4 : INTERFACES

Q8. An Interface defines rules that a class must follow. it specifies methods without implementations, ensuring consistency accaross different classes.
The ~~'Student'~~ 'Registrable' defined the 'register Course ()' function without implementation.

Q9. Implementing an interface forces a class to define all required methods. This ensures reliability and enforces a standard structure in the program.
The 'student' class implemented 'Registrable' and provided the 'Register Course' method.

PART 5 : MIXINS

Q10. A Mixin is used to add reusable functionality to a class without inheritance. it allows sharing behavior accross multiple classes. The 'Attendance Mixin' stored attendance and provided a method to increase it.

Q11. Mixins add behavior to a class without inheritance. The 'attendance Mixin' was applied to 'Student', allowing attendance to be marked three times and printed.

PART 6 : COLLECTIONS

Q12. A List is an ordered collection used to store multiple objects. It is useful for managing and iterating over grouped data like students. A List was created to store three 'Student' objects and loop through them.

Q13. A map stores data as key-value pairs. It is useful when data needs to be accessed quickly using a unique key, such as a student ID.

PART 7 : ANONYMOUS And ARROW FUNCTIONS

Q14. Anonymous functions are functions without names used for short, temporary tasks. They are commonly used inside loops and callbacks for cleaner code. An anonymous function was used inside 'forEach' to print student names from a list.

Q15. Arrow functions provide a short and simple syntax for single-expression functions. They make the code concise and easy to read. An arrow function was used to print a greeting for a student.

PART 8 : Asynchronous PROGRAMMING

Q16. Async functions allows tasks that take time to run without stopping the program. Waiting simulates real-world operations like loading data from a server. The 'Load Students ()' function waited for two seconds before returning a list of students.

Q17. Using 'await' ensures the program waits for async tasks to finish before continuing. Async programming helps apps remain responsive and efficient. In 'main()', 'loadStudents()' was awaited and the number of students loaded was printed.

PART 9: INTEGRATION CHALLENGE

Q18. Mixins are useful for adding specific behavior without creating a parent-child relationship. Inheritance was used for structure, while mixins were used to add features like attendance and notifications.

Q19. The new mixin 'NotificationMixin' was used to print a message when a student registered for a course. It was applied to the 'Student' class to add notifications without changing existing logic.

Q20. Learning Dart helps understand Flutter because flutter uses Dart for app logic. Dart concepts directly apply to Flutter development, making app creation easier and more structured.