Dart Programming Lab 1

Part 1

Q1. A function is a reusable block of code designed to perform a specific task. Instead of writing same codes many times, a function allows us to write it once and use it whenever needed by just calling that function by name.

Reasons why we created this function is that if we need to change the message in the future, it can be updated in one place, making the program easier to maintain, read and reuse.

Q2. Named parameters allow us to pass arguments to a function by explicitly specifying the parameter names rather than relying on their position.

Named parameters are helpful as they improve readability, reduces errors, makes function more flexible especially when dealing with optional and multiple parameters

Q3 Optional parameters allow us to make some function parameters not mandatory when the function is being called.

This can happen in two different ways which are by optional positional parameters where a parameter is placed in square brackets ([ ]) and optional

named parameters where parameter is placed in Curly braces ({ }).
With these two different ways optional parameters are given default values and if not they are null.

Part 2

Q4. Constructors are special methods that are used to initialize a new object from a class.
they are important because they ensure that an object starts in a valid state with all the necessary data.

Q5. An object is an instance of a class.
We create objects to use the blueprint (class) to store actual data and perform actions defined in that class.

Part 3

Q6. A class is a blueprint that defines the properties (variables and behaviors / methods that an object instantiated from it will have.

Q7. Inheritance allows one class to reuse and take on properties and methods of another class.

Student class reuses Person class by inheriting its name property and introduce method avoiding duplication

Part 4

Q8. Interface is like a contract that defines what a class must implement without defining how ~~to~~ do it.

Interface contains methods which are not implemented so that each and every class that implements ~~will~~ must implement all those methods.

Q9. By implementing Registrable interface, The Student class is forced to provide its own logic or override registerCourse.

This enforces rules and ensures that diffrent classes can be treated similarly if they share the same interface.

Part 5

Q10. Mixins are powerful feature in Dart that allows us to reuse code across multiple class hierarchies without using inheritance.

Q11. Mixins add behavior to a class without changing its inheritance hierarchy Unlike inheritance with a forcing strict "is a", relationship.

Part 6

Q12. Lists are ordered data structures that are used when You need to store multiple objects of the same type in the specific sequence that can be indexed

Q13 Maps are simple key/value pairs. Keys and values in a map may be of any type. A map is a dynamic collection. In other words maps can grow and shrink at runtime.

Maps are useful for looking up data quickly using a unique identifier such as a student object by their ID.

Part 7

Q14, Anonymous function is a function without a name. They are often passed as arguments / Callbacks in other functions when you only want to use the logic at that ~~spot~~ specific spot.

Q15, Arrow function is a shorthand syntax (=>) for functions that contain only one expression. They improve simplicity and readability by removing the need for curly braces and return keywords.

Part 8

Q16. Async function is a function that allows us to perform a long-running task without freezing the entire application.

The waiting is brought by await keyword which pauses the function and wait for the task to finish execution.

Q17. In real app, async programming is vital for fetching data from a database or Internet API. It ensures the user interface remains responsive while data is loading in the background.

Part 9

Q18. (inheritance creates a strict vertical hierarchy (a student is a Person), while mixins provide horizontal code sharing (a student can attendance).
Mixins are very useful because a class can use many mixins but only inherit from one superclass.

Q19. The new mixin NotificationMixin, ~~we~~ we applied it to Student to automatically trigger a notification whenever the registerCourse action occurs, demonstrating how behavior can be dynamically added.

Q20. After from the research we did, we found that Flutter is entirely built on Dart. So understanding how dart works is the foundation for building any function mobile app with Flutter.