

Programming Tasks

1. Collections, String Manipulation, and Functions

Write a program that:

- ****Accepts a list of sentences:****

```
dart  
  
List<String> sentences = [  
    "Dart is awesome",  
    "Flutter is amazing",  
    "I love programming"  
];  
...
```
 - ****Defines a function `countWords(List<String> sentences)` that:****
 - Counts the total number of words across all sentences.
 - Returns the count.
 - Prints the total word count and the longest word in the list.
-

2. Classes, Inheritance, and Static Methods

Write a program that:

- ****Creates a base class `Vehicle` with:****
 - Properties: `String brand` and `int year`.
 - A method `describe()` that prints the brand and year.

- **Creates a subclass `Car` that:**
 - Extends `Vehicle`.
 - Adds a property `int mileage`.
 - Overrides the `describe()` method to include mileage.
 - Adds a static method `Car.totalMileage(List<Car> cars)` to calculate and print the total mileage of a list of cars.
- **In `main()`, create a list of `Car` objects, calculate the total mileage, and print the details of each car.**

3. Nested Loops, Conditional Logic, and String Manipulation

Write a program that:

- **Prints a multiplication table from 1 to 5 using nested loops.**
- **Skips printing results where either multiplier or multiplicand is even.**
- **Formats the output so each result is padded to 4 characters for better alignment.**

Example Output:

```dart

```
1 3 5
3 9 15
5 15 25
```

...

---

## ## 4. Null Safety, Optional Parameters, and Logical Operators

Write a program that:

- **Defines a class `User` with:**
  - Properties: `String? name`, `int? age`, and `String? email`.
- **A method `isAdult()` that:**
  - Returns `true` if the age is 18 or older; otherwise, returns `false`.
  - Checks for null values using the null-aware operator (`??`).
- **In `main()`, create a `User` object with only the name and email fields, leaving age as null.**
- **Print whether the user is an adult or not.**

---

## ## 5. Mixins, Abstract Classes, and Enum

Write a program that:

- **Defines an abstract class `Appliance` with:**
  - A method `turnOn()`.
  - An abstract method `operate()`.
- **Creates a mixin `TimerFeature` that:**
  - Adds a method `setTimer(int minutes)` that prints "Timer set for \$minutes minutes."
- **Defines an enum `ApplianceType` with values `WASHER`, `DRYER`, and `OVEN`.**
- **Creates a class `Washer` that:**
  - Extends `Appliance`.
  - Mixes in `TimerFeature`.
  - Implements `operate()` to print "Washing clothes...".
- **In `main()`, create an instance of `Washer`, turn it on, set a timer, and operate it.**

