Programming Tasks

1. Functions, Exception Handling, and Loops

Write a program that:

- Defines a function `divideNumbers(int a, int b)` that:
  - Tries to divide `a` by `b`.
  - Throws an exception if `b` is zero with the message "Cannot divide by zero!".
  - Catches the exception and prints the error message.
  - Returns the result if no exception is thrown.
- Calls the function with different inputs and prints the results.

2. Lists, Maps, and Conditional Statements

Write a program that:

- Takes a list of student scores:

  ```dart
  List<int> scores = [85, 72, 90, 66, 78];
  ```

- Maps the scores to grades using the following criteria:
  - 90+ = "A"
  - 80-89 = "B"
  - 70-79 = "C"
  - Below 70 = "D"
- Stores the results in a `Map<int, String>` where the key is the score, and the value is the grade.
- Prints the scores and their corresponding grades.

3. OOP, Constructors, and Named Parameters

Write a program that:

- Defines a class `Product` with:

  - Properties: `String name`, `double price`, `int stock`.

  - A constructor using named parameters with default values for `price` and `stock` (e.g., `price = 0.0`, `stock = 0`).

  - A method `sell(int quantity)` that reduces the stock and prints a confirmation message.

- In `main()`, create an instance of `Product`, sell a few units, and print the updated stock.

4. Mixins, Abstract Classes, and Method Overriding

Write a program that:

- Defines an abstract class `Employee` with:
  - A method `double calculateSalary()` (abstract).
- Creates a mixin `Bonus` that:
  - Adds a method `double addBonus(double baseSalary)` that increases the salary by 10%.
- Implements a class `Manager` that:
  - Extends `Employee`.
  - Mixes in `Bonus`.
  - Overrides `calculateSalary()` to calculate a base salary of 5000.
- In `main()`, create a `Manager` instance, calculate the salary, apply the bonus, and print the final salary.