Abstraction

1. **Shape Area Calculation**: Create an abstract class `Shape` with a method `area`. Implement it in subclasses `Circle` and `Rectangle`.
2. **Vehicle Abstraction**: Define an abstract class `Vehicle` with abstract methods `start` and `stop`. Implement it in `Car` and `Bike` classes.

Inheritance

3. **Animal Inheritance**: Create a base class `Animal` with a method `makeSound`. Derive classes `Dog` and `Cat` that override the method.
4. **Employee Hierarchy**: Create a base class `Employee` with a method `calculateSalary`. Derive classes `Manager` and `Developer` that override the method.

Polymorphism

5. **Shape Polymorphism**: Use polymorphism to create different shapes (e.g., `Circle`, `Square`) and calculate their areas.
6. **Payment Method**: Implement polymorphism to handle different payment methods (`CreditCard`, `Paypal`).

Encapsulation

7. **Bank Account**: Implement a `BankAccount` class that encapsulates the account balance and provides methods to deposit and withdraw money.
8. **Student Info**: Create a `Student` class that encapsulates student details (name, age, grades) and provides methods to update and retrieve them.

Interface

9. **Printable Interface**: Create an interface `Printable` with a method `print`. Implement it in classes `Book` and `Magazine`.
10. **Movable Interface**: Define an interface `Movable` with methods `moveForward` and `moveBackward`. Implement it in `Car` and `Robot`.

Method Overriding

11. **Animal Sounds**: Override the `makeSound` method in derived classes of `Animal` (`Dog`, `Cat`).
12. **Vehicle Start**: Override the `start` method in derived classes of `Vehicle` (`Car`, `Motorcycle`).

Method Overloading

13. **Math Operations**: Create a class `MathOperations` with overloaded methods `add` for adding two integers, two floats, and three integers.
14. **Print Overloading**: Create a class with overloaded `print` methods to print integers, floats, and strings.

Arrays

15. **Sum of Array**: Write a function to calculate the sum of all elements in an integer array.
16. **Find Maximum**: Write a function to find the maximum element in an array.
17. **Reverse Array**: Write a function to reverse the elements of an array.
18. **Merge Arrays**: Write a function to merge two arrays into a single array.

Maps

19. **Frequency Count**: Write a function that counts the frequency of each element in an array using a map.
20. **Student Grades**: Create a map of student names to grades and write functions to add, update, and retrieve grades.

Sets

21. **Unique Elements**: Write a function to return the unique elements from an array using a set.
22. **Set Operations**: Write functions to perform union, intersection, and difference of two sets.

Queue

23. **Queue Implementation**: Implement a queue using a list and provide enqueue and dequeue operations.
24. **Queue Operations**: Write functions to perform basic operations on a queue (enqueue, dequeue, peek, isEmpty).

Stack

25. **Stack Implementation**: Implement a stack using a list and provide push and pop operations.
26. **Balanced Parentheses**: Write a function to check if a string has balanced parentheses using a stack.

Basic Programming

27. **Hello World**: Write a Dart program that prints "Hello, World!".
28. **Sum of Two Numbers**: Write a Dart function that takes two numbers as input and returns their sum.
29. **Even or Odd**: Write a Dart function that checks if a given number is even or odd.
30. **Factorial**: Write a Dart function to find the factorial of a given number.

31. **Palindrome Check**: Write a Dart function that checks if a given string is a palindrome.
32. **Reverse String**: Write a Dart function that reverses a given string.
33. **Prime Number**: Write a Dart function that checks if a given number is prime.
34. **Fibonacci Sequence**: Write a Dart function that prints the first N numbers of the Fibonacci sequence.
35. **Sum of Digits**: Write a Dart function that calculates the sum of the digits of a given number.
36. **Largest of Three Numbers**: Write a Dart function to find the largest among three numbers.
37. **Simple Interest**: Write a Dart program to calculate the simple interest.
38. **Leap Year Check**: Write a Dart function to check if a given year is a leap year.
39. **Armstrong Number**: Write a Dart function to check if a given number is an Armstrong number.
40. **Area of Circle**: Write a Dart program to calculate the area of a circle given its radius.
41. **Swap Two Variables**: Write a Dart function to swap two variables without using a third variable.
42. **GCD of Two Numbers**: Write a Dart function to find the greatest common divisor (GCD) of two numbers.
43. **LCM of Two Numbers**: Write a Dart function to find the least common multiple (LCM) of two numbers.
44. **Counting Vowels and Consonants**: Write a Dart function to count the number of vowels and consonants in a string.
45. **Remove Duplicates from List**: Write a Dart function to remove duplicates from a list.
46. **Second Largest Element**: Write a Dart function to find the second largest element in an array.
47. **Check Anagram**: Write a Dart function to check if two strings are anagrams.
48. **Sum of Array Elements**: Write a Dart function to find the sum of all elements in an array.
49. **Binary to Decimal**: Write a Dart function to convert a binary number to a decimal number.
50. **Decimal to Binary**: Write a Dart function to convert a decimal number to a binary number.