1. Create a class representing a "Book" with attributes such as title, author, and price. Implement a parameterized constructor to initialize these attributes and methods to display book details.

2. Develop a program to demonstrate constructor overloading by creating multiple constructors for a class.

3. Design a class hierarchy for different types of vehicles (e.g., Car, Bike, Truck) with common attributes like make, model, and year. Implement inheritance to avoid code duplication. Include a superclass "Vehicle" with methods like start() and stop(). Override these methods in subclasses to provide specific implementations for each type of vehicle.

4. Create an interface "Shape" with a method calculateArea(). Implement this interface with classes representing different geometric shapes (e.g., Circle, Rectangle, Triangle). Use polymorphism to calculate and display the area of each shape.

5. Design a class representing a "BankAccount" with private attributes such as account number, balance, and account holder name. Use encapsulation to provide public methods for deposit, withdrawal, and displaying account details while ensuring data integrity.

6. Write a Java program which will contain the user-defined package Calculator with all 4 basic arithmetic operations in a class and another class in package will contain operations like Square and Square Root (use `Math.sqrt()) method.`

7. Write a Java program to calculate the area and perimeter of a rectangle.

8. Create a program to find the factorial of a given number using iterative approaches.

9. Implement a simple calculator program that performs basic arithmetic operations (addition, subtraction, multiplication, division).

10. Implement a program that reads a sequence of integers from the user until a negative number is entered, then calculates and displays the sum and average of the entered numbers.

11. Write a Java program to check whether a given year is a leap year or not.

12. Create a program to find the largest and smallest elements in an array of integers.

13. Write a program to find the sum of elements in a 2D array and calculate the average.

14. Implement a method to check whether a given number is prime or not, and use it to find all prime numbers within a given range.

15. Using static method with the implementation of constructor write the program to implement counter.