```
import java.util.Scanner;
public class palindrome {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();
        scanner.close();
        // Step 2: Check if the input string is a palindrome
        if (isPalindrome(inputString)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
    }
}
// Function to check if a string is a palindrome
private static boolean isPalindrome(String str) {
    // Step 3: Remove spaces and convert to lowercase for case-insensitive comparison
    String cleanStr = str.replaceAll("\\s", "").toLowerCase();
    // Step 4: Compare characters from the beginning and end towards the center
    int left = 0;
    int right = cleanStr.length() - 1;
    while (left < right) {
        if (cleanStr.charAt(left) != cleanStr.charAt(right)) {
            return false; // If characters do not match, it's not a palindrome
            }
            left++;
            right--;
        }
        return true; // If all characters match, it's a palindrome
    }
}
```

```
public class RectangleAreaCalculator {

    // Method 1: Calculate area with length and width
    public static double calculateArea(double length, double width) {
        return length * width;
    }

    // Method 2: Calculate area with diagonal (assuming it's a rectangle)
    public static double calculateArea(double diagonal) {
        // Assuming the rectangle is a square, calculate the side length using diagonal
        double sideLength = diagonal / Math.sqrt(2);
        // Calculate area using the side length
        return sideLength * sideLength;
    }

    public static void main(String[] args) {
        // Test Method 1
        double area1 = calculateArea(5.0, 3.0);
        System.out.println("Area with length and width: " + area1);

        // Test Method 2
        double area2 = calculateArea(7.0);
        System.out.println("Area with diagonal: " + area2);
    }
}
```

```
public class Person {

    private String name;
    private int age;

    // Parameterized constructor
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Method to print details
    public void printDetails() {
        System.out.println("Person Details:");
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }

    public static void main(String[] args) {
        // Instantiate an object of the Person class
        Person person1 = new Person("John Doe", 25);

        // Print details of the person
        person1.printDetails();
    }
}
```

```
public class Book {

    private String title;
    private int pages;

    // Default constructor
    public Book() {
        this.title = "unknown";
        this.pages = 0;
    }

    // Method to print details
    public void printDetails() {
        System.out.println("Book Details:");
        System.out.println("Title: " + title);
        System.out.println("Pages: " + pages);
    }

    public static void main(String[] args) {
        // Instantiate an object of the Book class using the default constructor
        Book book1 = new Book();

        // Print details of the book
        book1.printDetails();
    }
}
```

```
public class Rectangle {

    private double length;
    private double width;

    // Parameterized constructor
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    // Method to calculate and print the area
    public void calculateAndPrintArea() {
        double area = length * width;
        System.out.println("Area of the rectangle: " + area);
    }

    public static void main(String[] args) {
        // Instantiate an object of the Rectangle class with length and width
        Rectangle rectangle1 = new Rectangle(5.0, 3.0);

        // Calculate and print the area of the rectangle
        rectangle1.calculateAndPrintArea();
    }
}
```

```
public class Cars {

    private String model;
    private int year;

    // Default constructor
    public Cars() {
        this.model = "unknown";
        this.year = 0;
    }

// Method to print details
public void printDetails() {
    System.out.println("Car Details:");
    System.out.println("Model: " + model);
    System.out.println("Year: " + year);
}

public static void main(String[] args) {
    // Instantiate an object of the Car class using the default constructor
    Cars car1 = new Cars();

    // Print details of the car
    car1.printDetails();
    }
}
```

```
import java.util.ArrayList;
class Account {
    private String accountNumber;
    private double balance;

    public Account(String accountNumber) {
        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. New balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount. Please enter a positive value.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient funds.");
        }
    }

    public double getBalance() {
        return balance;
    }
}

class Customer {
```

```
    private String name;
    private String customerId;
    private Account account;

    public Customer(String name, String customerId, Account account) {
        this.name = name;
        this.customerId = customerId;
        this.account = account;
    }

    public void deposit(double amount) {
        account.deposit(amount);
    }

    public void withdraw(double amount) {
        account.withdraw(amount);
    }

    public double checkBalance() {
        return account.getBalance();
    }
}

public class Transaction {
    public static void main(String[] args) {
        // Create an account for a customer
        Account account1 = new Account("123456789");

        // Create a customer
        Customer customer1 = new Customer("John Doe", "CUST123", account1);

        // Perform transactions
        customer1.deposit(1000.0);
        customer1.withdraw(500.0);
```

```
        // Check balance
        double balance = customer1.checkBalance();
        System.out.println("Current balance: " + balance);
    }
}
```

```
class Vehicle {
    private String brand;
    private int year;

    public Vehicle(String brand, int year) {
        this.brand = brand;
        this.year = year;
    }

    public void displayInfo() {
        System.out.println("Brand: " + brand);
        System.out.println("Year: " + year);
    }
}

// Derived class - Car (Single Inheritance)
class Car extends Vehicle {
    private int numberOfDoors;

    public Car(String brand, int year, int numberOfDoors) {
        super(brand, year);
        this.numberOfDoors = numberOfDoors;
    }

    public void displayCarInfo() {
        System.out.println("Car Information:");
        displayInfo(); // Calling method from the base class
        System.out.println("Number of Doors: " + numberOfDoors);
    }
}

// Derived class - Motorcycle (Single Inheritance)
class Motorcycle extends Vehicle {
    private String type;
```

```
    public Motorcycle(String brand, int year, String type) {
        super(brand, year);
        this.type = type;
    }

    public void displayMotorcycleInfo() {
        System.out.println("Motorcycle Information:");
        displayInfo(); // Calling method from the base class
        System.out.println("Type: " + type);
    }
}

// Derived class - Bicycle (Hierarchical Inheritance)
class Bicycle extends Vehicle {
    private String style;

    public Bicycle(String brand, int year, String style) {
        super(brand, year);
        this.style = style;
    }

    public void displayBicycleInfo() {
        System.out.println("Bicycle Information:");
        displayInfo(); // Calling method from the base class
        System.out.println("Style: " + style);
    }
}

public class VehicleHierarchyExample {
    public static void main(String[] args) {
        // Create objects for each type of vehicle
        Car car = new Car("Toyota", 2022, 4);
        Motorcycle motorcycle = new Motorcycle("Harley Davidson", 2021, "Cruiser");
```

```
        Bicycle bicycle = new Bicycle("Giant", 2020, "Mountain");

        // Display information for each vehicle
        System.out.println("----- Car Information -----");
        car.displayCarInfo();
        System.out.println("\n----- Motorcycle Information -----");
        motorcycle.displayMotorcycleInfo();
        System.out.println("\n----- Bicycle Information -----");
        bicycle.displayBicycleInfo();
    }
}
```