**Exercise 1:**

**Create an interface Drawable with a method draw(). Implement this interface in classes Circle and Square. Explain the concept of interfaces and how they are used in this example**.

**Exercise 2:**

**Create an interface Playable with a method playSound(). Implement this interface in classes Piano and Guitar. Explain the benefits of using interfaces for implementing common behavior.**

**Exercise 3:**

**Create an interface Shape with methods calculateArea() and calculatePerimeter(). Implement this interface in classes Circle and Rectangle. Explain how interfaces allow multiple inheritance**.

**Exercise 4:**

**Create an interface Movable with methods moveLeft(), moveRight(), moveUp(), and moveDown(). Implement this interface in a class Player. Explain how interfaces help in achieving a consistent API for different classes.**

**Exercise 5:**

**Create an interface Logger with a method log(String message). Implement this interface in classes ConsoleLogger and FileLogger. Demonstrate how you can achieve polymorphism using interfaces.**

**Exercise 6:**

**Create an interface Database with methods connect() and disconnect(). Implement this interface in classes MySQLDatabase and OracleDatabase. Illustrate how interfaces can be used to switch between different implementations seamlessly**.

**Exercise 7:**

**Create an interface Resizable with a method resize(double factor). Implement this interface in a class ResizableCircle. Explain how interfaces can introduce new behavior to existing classes.**

**Exercise 8:**

**Create an interface ElectricDevice with a method turnOn() and turnOff(). Implement this interface in classes Fan and Light. Explain how interfaces allow you to define a contract for unrelated classes.**

**Exercise 9:**

**Create an interface PaymentProcessor with methods processPayment(double amount) and refundPayment(double amount). Implement this interface in classes CreditCardProcessor and PayPalProcessor. Explain how interfaces promote loose coupling.**

**Exercise 10:**

**Create an interface Animal with a method makeSound(). Implement this interface in classes Dog and Cat. Explain how interfaces facilitate code organization and polymorphism.**

**Exercise 11:**

**Create an interface Employee with methods calculateSalary() and displayInfo(). Implement this interface in classes Manager and Developer. Explain how interfaces enforce certain behavior across different classes.**

**Exercise 12:**

**Create an interface Playable with methods play(), pause(), and stop(). Implement this interface in classes AudioPlayer and VideoPlayer. Illustrate how interfaces can be used to create a unified interface for different types of media players.**

**Exercise 13:**

**Create an interface BankAccount with methods deposit(double amount) and withdraw(double amount). Implement this interface in classes SavingsAccount and CheckingAccount. Explain how interfaces help in achieving code reusability and polymorphism.**

**Exercise 14:**

**Create an interface Comparable with a method compareTo(Object other). Implement this interface in a class Person. Demonstrate how you can use the compareTo() method to compare instances of Person.**

**Exercise 15:**

**Create an interface Resizable with methods resizeWidth(double factor) and resizeHeight(double factor). Implement this interface in a class ResizableRectangle. Illustrate how interfaces can help in designing flexible and extensible classes.**

**Exercise 16:**

**Create an interface Taxable with a method calculateTax(double income). Implement this interface in classes Individual and Business. Explain how interfaces allow you to provide a common calculation method for different entities.**

**Exercise 17:**

**Create an interface Shape with methods calculateArea() and calculatePerimeter(). Implement this interface in classes Circle and Rectangle. Demonstrate how interfaces can be used to enforce consistent behavior across a hierarchy of classes.**

**Exercise 18:**

**Create an interface Cooking with a method prepareFood(). Implement this interface in classes Chef and HomeCook. Explain how interfaces allow different classes to provide unique implementations while adhering to a common contract.**

**Exercise 19:**

**Create an interface Displayable with methods show(). Implement this interface in classes Image and Text. Explain how interfaces can be used to handle different types of data in a unified way.**

**Exercise 20:**

**Create an interface Authentication with methods login(String username, String password) and logout(). Implement this interface in classes User and Admin. Explain how interfaces can enforce security-related behavior across different user roles.**