Java Method Overloading

Exercise 1: Area Calculation

Create a class AreaCalculator with overloaded methods named area to calculate:

1. The area of a **circle** (accepts radius as parameter).
2. The area of a **rectangle** (accepts length and width).
3. The area of a **square** (accepts side).
4. The area of a **triangle** (accepts base and height).

Exercise 2: Maximum Finder

Write a class MaxFinder with overloaded methods max to find the maximum:

1. Between **two integers**.
2. Between **two doubles**.
3. Between **three integers**.

Exercise 3: String Repeater

Create a class StringPrinter with an overloaded method print that:

1. Prints a **string once**.
2. Prints a **string multiple times** (accept string and count).
3. Prints a **string in uppercase** (accept string and a boolean flag).

Exercise 4: Simple Calculator

Write a class Calculator with overloaded methods add that:

1. Adds **two integers**.
2. Adds **two doubles**.
3. Adds **three integers**.
4. Concatenates **two strings**.

Exercise 5: Bank Interest Calculator

Create a class Bank with overloaded methods calculateInterest:

1. Accepts **principal and time** (assume fixed rate).
2. Accepts **principal, time, and rate**.
3. Accepts **principal, time, rate, and compounding frequency** (compound interest).

Exercise 6: Distance Converter (Advanced)

Create a class DistanceConverter with overloaded methods convert that:

1. Converts **kilometers to meters**.
2. Converts **meters to kilometers**.
3. Converts **miles to kilometers**.
4. Converts **kilometers to miles**.

Exercise 7: Ticket Booking System

Create a class TicketBooking with overloaded bookTicket:

1. Accepts only **name** → default economy class.
2. Accepts **name and seat type** (Economy/Business).
3. Accepts **name, seat type, and meal preference**.
4. Accepts **name, seat type, meal preference, and number of tickets**.

Exercise 8: Method Overloading with Type Promotion

Write a class OverloadTest with overloaded show methods:

1. Accepts an int.
2. Accepts a long.
3. Accepts a float.
4. Accepts a double.

Test what happens if you pass values like 5, 5L, 5.0f, and 5.0.

Constructor Overloading in Java

Exercise 1: Rectangle

Create a class Rectangle with overloaded constructors:

1. No-argument constructor → sets length = 1, width = 1.
2. Constructor with one argument → sets both length and width to the same value (square).
3. Constructor with two arguments → sets length and width to different values.

Add a method area() to calculate the area.

Exercise 2: Student Information

Create a class Student with overloaded constructors:

1. Accepts only name.
2. Accepts name and age.
3. Accepts name, age, and department.

Add a method displayInfo() to print student details.

Exercise 3: Bank Account

Create a class BankAccount with overloaded constructors:

1. No-argument constructor → initializes account with default balance = 0.
2. Constructor with accountNumber.
3. Constructor with accountNumber and balance.
4. Constructor with accountNumber, balance, and accountHolderName.

Exercise 4: Book

Create a class Book with overloaded constructors:

1. Accepts only title.
2. Accepts title and author.
3. Accepts title, author, and price.
4. Accepts title, author, price, and year.

Add a method displayBook() to show details.

Exercise 5: Employee Salary

Create a class Employee with overloaded constructors:

1. Accepts only name.
2. Accepts name and salary.
3. Accepts name, salary, and designation.

Add a method showDetails() to print employee info.

Exercise 6: Car

Create a class Car with overloaded constructors:

1. No-argument constructor → default brand = "Unknown", model = "Unknown".
2. Constructor with brand.
3. Constructor with brand and model.
4. Constructor with brand, model, and year.

Add a method displayCar() to show car details.

Exercise 7: Complex Number

Create a class Complex with overloaded constructors:

1. No-argument constructor → initializes real = 0, imaginary = 0.
2. Constructor with one argument → initializes real with that value and imaginary = 0.
3. Constructor with two arguments → initializes both real and imaginary parts.

Add a method display() to print the complex number in form a + bi.

Exercise 8: Library Member

Create a class Member with overloaded constructors:

1. Constructor with memberId.
2. Constructor with memberId and name.
3. Constructor with memberId, name, and membershipType.
4. Constructor with memberId, name, membershipType, and joiningDate.

Exercise 9: Point in 2D Space

Create a class Point with overloaded constructors:

1. No-argument constructor → initializes (0,0).
2. Constructor with x.
3. Constructor with x and y.

Add a method distanceFromOrigin() that returns the distance of the point from (0,0).

Exercise 10 (Challenge): Constructor Chaining

Create a class Laptop with overloaded constructors that use **constructor chaining** (this() keyword):

1. No-argument constructor → default brand = "Unknown", RAM = 4 GB.
2. Constructor with brand.
3. Constructor with brand and RAM.
4. Constructor with brand, RAM, and price.

Add a method showSpecs() to print laptop details.