

# Beginner-Level Java Practice Problems

## 1. Class & Object (Intro)

Create a class `Book` with attributes `title` and `author`. Write a program to create two `Book` objects and print their details using a method `displayInfo()`.

---

## 2. Object Reference

Create a class `Student` with attributes `name` and `rollNumber`. Assign one object reference to another variable and show that changes made using one reference are reflected in the other.

---

## 3. Static Variable

Create a class `Car` with attributes `model` and a static variable `totalCars`. Each time a new `Car` object is created, increase `totalCars` by 1. Print the total number of cars created.

---

## 4. Static Method

Create a class `MathUtility` with a static method `findSquare(int n)` that returns the square of a number. Demonstrate calling this method without creating an object.

---

## 5. Final Variable

Create a class `Circle` with a `final` variable `PI = 3.14159`. Ask the user for the radius and calculate the area of the circle. Show that `PI` cannot be reassigned.

---

## 6. Initialization Block

Create a class `Employee` where an initialization block sets a default department name. Allow users to enter employee details (`name`, `id`) and print them along with the default department.

---

## 7. Static Block

Create a class `AppConfig` with a static block that prints “Application is starting...” before the `main` method runs.

---

## 8. Constructors (Default & Parameterized)

Create a class `Person` with attributes `name` and `age`. Write a default constructor that sets values to "Unknown" and 0. Write a parameterized constructor to set custom values. Demonstrate both.

---

## 9. Constructor Overloading

Create a class `Rectangle` with attributes `length` and `width`. Provide constructors:

- Default constructor → sets both to 1
  - One-parameter constructor → square
  - Two-parameter constructor → rectangle  
Print area for all three cases.
- 

## 10. 1D Array

Write a program that asks the user to enter marks of 5 students in a 1D array and prints the highest mark using a loop.

---

## 11. 2D Array

Write a program to store marks of 3 students in 3 subjects using a 2D array. Print the average marks of each student.

---

## 12. Jagged Array

Create a jagged array to store the marks of students in different subjects (e.g., Student1 has 2 subjects, Student2 has 3 subjects, Student3 has 4 subjects). Print all marks.

---

## 13. Primitive vs Reference Type Array

Create two arrays:

- `int[] numbers = {1, 2, 3};`
  - `String[] names = {"Alice", "Bob", "Charlie"};`  
Modify one element in each array and show how primitive vs reference behaves differently.
- 

## 14. Scanner (User Input)

Ask the user to enter their name, age, and favorite number. Print a greeting message using their details.

---

## 15. Scanner + Array

Ask the user to enter the number of students, then read all their names into an array and print them.

---

## 16. Math Class (Random Numbers)

Generate a random number between 1 and 100 using `Math.random()`. Ask the user to guess the number. Give hints like “Too High” or “Too Low” until they guess correctly.

---

## 17. Math Class (Functions)

Ask the user for two numbers. Use `Math.max()`, `Math.min()`, `Math.pow()`, and `Math.sqrt()` to display useful results.

---

## 18. Class + Array of Objects

Create a class `Movie` with attributes `title` and `rating`. Ask the user to enter 3 movies, store them in an array, and print all movie details.

---

## 19. Combining Concepts (Static + Object)

Create a class `BankAccount` with attributes `accountNumber` and `balance`. Use a static variable `totalAccounts` to track how many accounts were created. Print details of accounts and total accounts created.

---

## 20. Challenge (Multiple Topics Together)

Create a class `StudentRecord` with attributes `name`, `roll`, and `marks[ ]` (array).

- Use a constructor to initialize name and roll.
- Use `Scanner` to input marks for 3 subjects.
- Use an initialization block to set a default college name.
- Use a static method to calculate average marks.
- Print student details and average marks.