**Additional Theory Questions**

1. Explain the difference between **instance variables** and **static variables** with examples.
2. What happens if you don’t write any constructor in a class? Give an example.
3. Can a constructor be **overloaded**? Justify with an example.
4. Why can’t we overload a method only by changing its return type?
5. What is the difference between a **copy constructor** and a **clone() method** in Java?
6. Explain **memory allocation** for objects in heap vs. stack with a diagram.
7. What is the difference between **default, private, protected, public** access specifiers? Give one use case for each.
8. Compare **static blocks** vs **constructors** in Java. When is each executed?
9. Arrays in Java are objects. Prove this statement.
10. What exceptions are commonly thrown while working with arrays in Java? Explain with examples.

**Additional Coding/Program Questions**

1. **Class & Object**
   * Write a program to represent a **Library Book** with attributes title, author, price. Create multiple objects and display their details.
2. **Constructors**
   * Write a program to demonstrate **constructor chaining** using this() keyword.
   * Write a program to count how many objects of a class have been created using a static variable inside constructors.
3. **Method Overloading**
   * Create a Calculator class with overloaded methods to calculate the **maximum of two, three, and four numbers**.
   * Create a class PrintData with overloaded methods print() that can print **int, double, char, and String** types.
4. **Static Members**
   * Write a program with a **Student class** having a static variable totalStudents. Each time an object is created, increment it and print total count.
   * Write a program to show that **static methods cannot access non-static variables directly**.
5. **Arrays**
   * Write a program to find the **second largest element** in a 1-D integer array.
   * Write a program to **reverse a string array** and display it.
   * Write a program to find the **transpose of a 2-D matrix**.
   * Use Arrays.sort() to sort an array of String objects in alphabetical order.
   * Demonstrate the difference between **Arrays.equals()** and == operator when comparing arrays.

**Higher-Order / Application-Oriented Questions**

1. Design a class BankAccount with methods deposit(), withdraw(), and a static method setInterestRate(). Demonstrate how the static method affects all objects.
2. Implement a **Student Report System** where marks of 5 subjects are stored in an array inside each object, and a method calculates the average.
3. Write a program to demonstrate **shallow copy vs deep copy** of an object containing arrays.
4. Write a program to simulate a **shopping cart** where products are stored in an array. Include methods to add, remove, and calculate total.
5. Create a program that demonstrates **method overloading** for calculating the area of geometric shapes, but include an additional case: **parallelogram and trapezium**.