

## Question 1: Java Collection Framework

Write a Java program that uses an ArrayList to store a list of student names. Implement a method to add a name, remove a name, and display all names in alphabetical order. Handle the case where the list is empty using a custom exception.

### Expected Output Example:

Adding names: Alice, Bob, Charlie  
Sorted names: Alice, Bob, Charlie  
Removing Bob...  
Sorted names after removal: Alice, Charlie  
Attempting to display empty list...  
Error: Student list is empty!

## Question 2: HashMap

Create a Java program that uses a HashMap to store employee IDs (Integer) as keys and their names (String) as values. Write methods to:

1. Add an employee to the map.
2. Retrieve an employee's name by ID.
3. Throw a custom exception if the ID is not found.
4. Display all employees in the format "ID: Name".

### Expected Output Example:

Adding employees...  
Employee Map: {101=John, 102=Jane, 103=Mike}  
Name for ID 102: Jane  
Name for ID 999: Error: Employee ID not found!

### Question 3: Exception Handling

Write a Java program that reads a list of integers from the user into an ArrayList. Calculate the average of the numbers. Handle the following exceptions:

1. ArithmeticException if the list is empty (to avoid division by zero).
2. Use a try-catch block to manage these exceptions and provide meaningful error messages.

#### Expected Output Example:

```
Enter numbers (type 'done' to finish):
45
23
abc
Error: Invalid input, please enter a valid integer.
67
done
Average: 45.0
```

### Question 4: HashMap and Exception Handling

Write a Java program that uses a HashMap to store product IDs (String) and their prices (Double). Implement a method to apply a discount to a product's price based on a percentage input. Handle the following cases:

1. Throw a custom exception if the product ID doesn't exist.
2. Throw an IllegalArgumentException if the discount percentage is negative or greater than 100.
3. Update the price in the HashMap after applying the discount.

#### Expected Output Example:

```
Adding products: P001=$50.0, P002=$100.0
Applying 20% discount to P001...
New price for P001: $40.0
Applying 150% discount to P002...
Error: Discount percentage must be between 0 and 100!
Applying discount to P999...
Error: Product ID not found!
```