**Q1: Use Collectors API and Immutable Collections**

import java.util.\*;

import java.util.stream.Collectors;

public class CollectorsExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<String> names = new ArrayList<>();

System.out.println("Enter student names (type 'end' to finish):");

while (true) {

String input = scanner.nextLine();

if (input.equalsIgnoreCase("end")) break;

names.add(input);

}

Set<String> nameSet = names.stream().collect(Collectors.toSet());

List<String> immutableList = List.copyOf(names);

long countA = names.stream().filter(name -> name.startsWith("A")).count();

System.out.println("Names Set: " + nameSet);

System.out.println("Immutable List: " + immutableList);

System.out.println("Names starting with A: " + countA);

}

}

**Q2: Use var Keyword and File API**

import java.io.IOException;

import java.nio.file.\*;

import java.util.List;

public class FileReadUppercase {

public static void main(String[] args) {

try {

var path = Paths.get("input.txt"); // Ensure this file exists

var lines = Files.readAllLines(path);

for (var line : lines) {

System.out.println(line.toUpperCase());

}

}

catch (IOException e) {

System.out.println("Error reading file: " + e.getMessage());

}

}

}

**Q3: Switch Expressions & Text Blocks**

import java.util.Scanner;

public class SwitchTextBlockExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter day of the week (e.g., MONDAY): ");

String day = scanner.nextLine().toUpperCase();

String result = switch (day) {

case "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY" -> "Working Day";

case "SATURDAY", "SUNDAY" -> "Weekend";

default -> "Invalid";

};

String message = """

========= Result =========

You entered: %s

Classification: %s

=========================

""".formatted(day, result);

System.out.println(message);

}

}

**Q4: Java Records & Stream API**

import java.util.\*;

import java.util.stream.Collectors;

public class StudentRecordExample {

record Student(String name, int rollNumber, int marks) {}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<Student> students = new ArrayList<>();

System.out.print("How many students? ");

int n = Integer.parseInt(scanner.nextLine());

for (int i = 0; i < n; i++) {

System.out.print("Enter name: ");

String name = scanner.nextLine();

System.out.print("Enter roll number: ");

int roll = Integer.parseInt(scanner.nextLine());

System.out.print("Enter marks: ");

int marks = Integer.parseInt(scanner.nextLine());

students.add(new Student(name, roll, marks));

}

System.out.println("\nStudents with marks > 75:");

students.stream()

.filter(s -> s.marks() > 75)

.forEach(System.out::println);

System.out.println("\nStudents sorted by marks:");

students.stream()

.sorted(Comparator.comparingInt(Student::marks).reversed())

.forEach(System.out::println);

}

}

**Q5: Log4j2 Logging**

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

import java.io.IOException;

import java.nio.file.\*;

import java.util.List;

public class LoggingExample {

private static final Logger logger = LogManager.getLogger(LoggingExample.class);

public static void main(String[] args) {

try {

Path path = Paths.get("data.txt"); // Ensure this file exists

if (!Files.exists(path)) {

logger.warn("File does not exist. Operation skipped.");

return;

}

List<String> lines = Files.readAllLines(path);

logger.info("File successfully read. Number of lines: " + lines.size());

for (String line : lines) {

System.out.println(line);

}

}

catch (IOException e) {

logger.error("Exception while reading file", e);

}

}

}