**LAB 7**

***1.Student Record Management using ArrayList Write a Java program to: Create a class Student with id, name, and marks. Store multiple Student objects in an ArrayList. Display all student details. Search for a student by ID.***

import java.util.\*;

class Student {

    int id;

    String name;

    double marks;

    Student(int id, String name, double marks) {

        this.id = id;

        this.name = name;

        this.marks = marks;

    }

    void display() {

        System.out.println("ID: " + id + ", Name: " + name + ", Marks: " + marks);

    }

}

class StudentManagement {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        // Add students

        students.add(new Student(1, "Jack", 85.5));

        students.add(new Student(2, "Vijay", 78.0));

        students.add(new Student(3, "Sneha", 92.0));

        // Display all students

        System.out.println("All Student Details:");

        for (Student s : students) {

            s.display();

        }

        // Search by ID

        System.out.print("\nEnter student ID to search: ");

        int searchId = sc.nextInt();

        boolean found = false;

        for (Student s : students) {

            if (s.id == searchId) {

                System.out.println("Student found:");

                s.display();

                found = true;

                break;

            }

        }

        if (!found) {

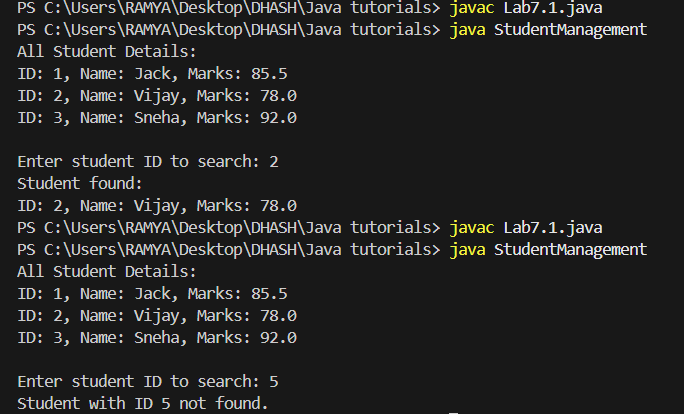
            System.out.println("Student with ID " + searchId + " not found.");

        }

        sc.close();

    }

}



***2: ArrayList Sorting and Updating Write a Java program to: Create an ArrayList of Strings (names of fruits). Add at least 5 names. Sort the list in ascending order. Replace one of the elements. Remove an element by index.***

import java.util.\*;

class FruitList {

    public static void main(String[] args) {

        ArrayList<String> fruits = new ArrayList<>();

        // Add fruit names

        fruits.add("Banana");

        fruits.add("Apple");

        fruits.add("Mango");

        fruits.add("Orange");

        fruits.add("Grapes");

        System.out.println("Original List: " + fruits);

        // Sort in ascending order

        Collections.sort(fruits);

        System.out.println("Sorted List: " + fruits);

        // Replace an element

        fruits.set(2, "Pineapple");  // Replace element at index 2

        System.out.println("After Replacement: " + fruits);

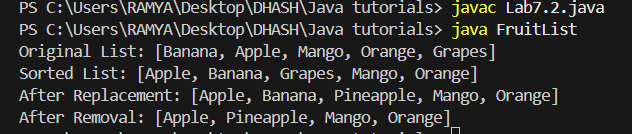
        // Remove by index

        fruits.remove(1);  // Remove element at index 1

        System.out.println("After Removal: " + fruits);

    }

}



***Q3: Even-Odd Separator Write a Java program to: Accept 10 integers from the user and store them in an ArrayList. Separate and store even and odd numbers into two different ArrayLists. Print all three lists: Original, Even, and Odd***

import java.util.\*;

public class EvenOddSeparator {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        ArrayList<Integer> original = new ArrayList<>();

        ArrayList<Integer> even = new ArrayList<>();

        ArrayList<Integer> odd = new ArrayList<>();

        System.out.println("Enter 10 integers:");

        for (int i = 0; i < 10; i++) {

            System.out.print("Number " + (i + 1) + ": ");

            int num = sc.nextInt();

            original.add(num);

            if (num % 2 == 0) {

                even.add(num);

            } else {

                odd.add(num);

            }

        }

        System.out.println("\nOriginal List: " + original);

        System.out.println("Even Numbers: " + even);

        System.out.println("Odd Numbers: " + odd);

        sc.close();

    }

}

