<u>**Experiment 4**</u>

Student Name: Rajat Katiyar   **UID: 22BCS15928**

Branch: BE-CSE   **Section/Group: KRG 2B**

Semester:6th   **Date of Performance: 12/02/2025**

Subject Name: Project Based Learning   **Subject Code: 22CSH-359**
              in Java with Lab

1. **Aim: Write a Program to perform the basic operations like insert, delete, display and search in list. List contains String object items where these operations are to be performed.**

2. **Objective: The objective of this program is to implement basic operations (insert, delete, display, and search) on a List containing String objects. The program will demonstrate how to manipulate a list using common list operations in Java, providing functionality to manage and interact with data stored in the list.**

3. Implementation/Code:

```java
import java.util.ArrayList;
import java.util.Scanner;

public class StringListOperations {

    private static ArrayList<String> list = new ArrayList<>();

    public static void insertItem(String item) {
        list.add(item);
    }

    public static void deleteItem(String item) {
        if (list.contains(item)) {
            list.remove(item);
            System.out.println(item + " has been removed.");
        } else {
            System.out.println(item + " not found in the list.");
        }
```

```java
    }
    public static void displayList() {
        if (list.isEmpty()) {
            System.out.println("The list is empty.");
```

```java
        } else {
            System.out.println("List items: " + list);
        }
    }

    public static void searchItem(String item) {
        if (list.contains(item)) {
            System.out.println(item + " is found in the list.");
        } else {
            System.out.println(item + " is not found in the list.");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\nSelect an operation:");
            System.out.println("1. Insert Item");
            System.out.println("2. Delete Item");
            System.out.println("3. Display List");
            System.out.println("4. Search Item");
            System.out.println("5. Exit");
            choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter item to insert: ");
                    String insertItem = sc.nextLine();
                    insertItem(insertItem);
                    break;
                case 2:
                    System.out.print("Enter item to delete: ");
                    String deleteItem = sc.nextLine();
                    deleteItem(deleteItem);
                    break;
```

```
                case 3:
                    displayList();
                    break;
                case 4:
                    System.out.print("Enter item to search: ");
                    String searchItem = sc.nextLine();
                    searchItem(searchItem);
                    break;
                case 5:
```

```
                    System.out.println("Exiting program.");
                    break;
                default:
                    System.out.println("Invalid choice! Please choose a valid option.");
            }
        } while (choice != 5);

        sc.close();
    }
}
```

4. Output:

```
3. Display List
4. Search Item
5. Exit
2

Enter item to delete: Apple
Apple has been removed.
```

```
Select an operation:
1. Insert Item
2. Delete Item
3. Display List
4. Search Item
5. Exit
3

The list is empty.
```

5. Learning Outcomes:

1. Learn how to perform basic **CRUD (Create, Read, Update, Delete)** operations on a **List** of **String** objects in Java.
2. Understand how to use the **ArrayList** class for dynamically storing and manipulating a collection of items.
3. Practice handling **user input** using the **Scanner** class for interaction with the program.
4. Implement methods for **searching**, **deleting**, and **displaying** items in a list efficiently.
5. Gain familiarity with **control flow** and **loops** to allow for continuous user interaction until the program is exited.