



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-3

Student Name: Abhigyan Singh
Branch: BE-CSE
Semester: 6th
Subject Name: PBLJ-Lab

UID: 22BCS14340
Section/Group: KRG_IOT-3/B
Date of Performance: 18/02/25
Subject Code: 22CSH-359

1. Aim:

The aim of this project is to develop a user-friendly Java application that allows users to efficiently manage a dynamic list of integers through a simple menu-driven interface. The application will enable users to perform basic operations such as adding, removing, displaying, and checking for the existence of elements in the list.

2. Objective:

The objectives of the Java program are to create a dynamic list using **ArrayList** for integer storage, develop a user-friendly menu-driven interface for navigation, and enable users to add and remove integers with appropriate confirmations and existence checks. The program will also display all current elements in the list, handle cases where the list is empty, provide functionality to check for the existence of specific integers, manage invalid inputs for a smooth user experience, and implement a clean exit option to terminate the program.

3. Implementation/Code:

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

public class Main {

    public static void main(String[] args) {

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

        Scanner scanner = new Scanner(System.in);

        while (true) { System.out.println();

            System.out.println(" -----");
            System.out.println("Menu:"); System.out.println("1.
Add element"); System.out.println("2. Remove
element"); System.out.println("3. Display
elements"); System.out.println("4. Check if element
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
exists"); System.out.println("5. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();
switch (choice) { case 1:

System.out.print("Enter element to add: "); int a =
scanner.nextInt();

list.add(a); System.out.println("Element added.");

break;

case 2:
System.out.print("Enter element to remove: "); int
removeElement = scanner.nextInt();

if (list.contains(removeElement)) {

list.remove(Integer.valueOf(removeElement));

System.out.println("Element removed.");

} else {
System.out.println("Element not found.");
}
break;

case 3:
if (list.isEmpty()) { System.out.println("The list is empty.");

} else {
System.out.println("List elements:"); for (int element :
list) {

System.out.println(element);
}
}
break; case 4:

System.out.print("Enter element to check: "); int
checkElement = scanner.nextInt();

if (list.contains(checkElement)) { System.out.println("Element exists in the
list.");

} else {
System.out.println("Element does not exist in the list.");
}
break; case 5:

System.out.println("Exiting program...");
```

```
scanner.close();

return;
default:
    System.out.println("Invalid choice, please try again.");
    }
}
```

4. Output:

```
-----
Menu:
1. Add element
2. Remove element
3. Display elements
4. Check if element exists
5. Exit
Enter your choice: 1
Enter element to add: 23
Element added.

-----
Menu:
1. Add element
2. Remove element
3. Display elements
4. Check if element exists
5. Exit
Enter your choice: 1
Enter element to add: 34
Element added.

-----
Menu:
1. Add element
2. Remove element
3. Display elements
4. Check if element exists
5. Exit
Enter your choice: 3
List elements:
23
34

-----
Menu:
1. Add element
2. Remove element
3. Display elements
4. Check if element exists
5. Exit
Enter your choice: 5
Exiting program...
```

5. Learning Outcome:

1. Learn object-oriented programming through classes and methods in Java.
2. Learn about the ArrayList.
3. Build interactive menu-driven console applicatio