

# ASSIGNMENT - 3

Name:- Ch.Mahanvitha

Reg.No:- 192211785

Course Code:- CSA0914

Course Name:- Programming  
in Java for Raspberry  
Pi

Faculty:- Dr. Hemavathi

# 1) ArrayList Operations.

Write a Java program to perform the following operations using an ArrayList:

- 1) Add elements to the list.
- 2) Remove an element by index.
- 3) Search for an element and print its position.
- 4) Iterate through the list and display all elements.

**Input:-** ArrayList Operations. menu:

- 1) Add Element
- 2) Remove Element by Index
- 3) Search for an Element
- 4) display All Elements
- 5) Exit.

1

Enter element to add: Apple

**Output:-** Element added.

**Pseudo Code:-**

START

CREATE an empty ArrayList named "list"

CREATE a Scanner object for user input

REPEAT

DISPLAY menu options:

- 1) Add Element
- 2) Remove Element by Index
- 3) Search for an Element
- 4) display All Elements
- 5) Exit

READ user choice.

IF choice is 1 THEN  
PROMPT user to enter an element  
ADD the element to the ArrayList  
DISPLAY "Element added"  
ELSE IF choice is 2 THEN  
PROMPT user to enter index to remove  
IF index is valid (within bounds of list)  
REMOVE the element at the specified index  
DISPLAY "Element removed"  
ELSE  
DISPLAY "Invalid index"  
END IF  
ELSE IF choice is 3 THEN  
PROMPT user to enter element to search  
SEARCH for the element in the ArrayList  
IF element is found THEN  
DISPLAY "Element found at index" and  
its position  
DISPLAY "Element not found"  
END IF  
ELSE IF choice is 4 THEN  
DISPLAY all elements in ArrayList  
ELSE  
DISPLAY "Invalid choice"  
END IF  
UNTIL user chooses to exit  
CLOSE the Scanner  
END

## Program:-

```
import java.util.ArrayList;
import java.util.Scanner;
public class ArrayListOperations {
    public static void main (String [ ] args) {
        Scanner scanner=new Scanner (System.in);
        ArrayList <String> list = new ArrayList <> ();
        while(true) {
            System.out.println ("InArrayList Operations
                                Menu:");
            System.out.println ("1. Add element");
            System.out.println ("2. Remove Element by
                                Index");
            System.out.println ("3. Search for an Element");
            System.out.println ("4. Display All Elements");
            System.out.println ("5. Exit");
            int choice=scanner.nextInt();
            scanner.nextLine ();
            switch (choice) {
                case 1:
                    System.out.print ("Enter element to
                        add: ");
                    String element=scanner.nextLine();
                    list.add(element);
                    System.out.println ("Element added");
                    break;
```

case 2:

```
System.out.print("Enter index to remove:");
int index = scanner.nextInt();
if (index >= 0 && index < list.size()) {
    list.remove(index);
    System.out.println("Element at index");
    break;
}
```

case 3:

```
System.out.print("Enter element to search:");
String searchElement = scanner.nextLine();
if (position != -1) {
    System.out.println("Element found at:");
} else {
    System.out.println("Element not found");
}
break;
default:
    System.out.println("Invalid");
}
```

2) Create a program that demonstrates the use of a HashSet to store a collection of names. Include the following features:

- Add a name to the set.
- Remove a name from the set.
- Check if a specific name is present in the set.
- Display all names.

**Input:-** HashSet operations:-

- 1) Add a Name
- 2) Remove a Name
- 3) Check if a Name is present
- 4) Display All Names
- 5) Exit.

Enter your choice: 1

Enter name to add: Alice.

**Output:-** Name added successfully.

**Pseudo Code:-**

START

CREATE an empty HashSet called "nameSet"

CREATE a Scanner object for user input.

REPEAT

DISPLAY menu options:

- 1) Add a Name
- 2) Remove a Name
- 3) Check if a Name is present
- 4) Display All Names

5) EXIT.

READ user choice

IF choice is 1 THEN

PROMPT user to enter a name to add

IF the name is not already in the set THEN

ADD the name to the HashSet

DISPLAY "Name added"

ELSE

DISPLAY "Name already exist"

ELSE END IF

• IF choice is 2 THEN

PROMPT user to enter a name to remove

IF the name exists in the set THEN

REMOVE name from HashSet

DISPLAY "Name removed"

ELSE

DISPLAY "Name not found"

ENDIF

UNTIL user chooses to exit

• CLOSE the Scanner

END.

Program:-

```
import java.util.HashSet;
```

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

```
public class NameHashSet {
```

```
public static void main (String [ ] args) {
    HashSet < String > nameset = new HashSet <> ();
    Scanner scanner = new Scanner (System.in);
    while (true) {
        System.out.println ("In --- HashSet --");
        System.out.println ("1. Add a Name");
        System.out.println ("2. Remove a Name");
        System.out.println ("3 Check Name");
        System.out.println ("4. Display Names");
        System.out.println ("5. Exit");
        System.out.println ("Enter choice:");
        int choice = scanner.nextInt();
        scanner.nextLine();
        switch (choice) {
            case 1:
                S.O.P ("Enter name to add:");
                String nameToAdd = sc.nextLine();
                break;
            case 2:
                S.O.P ("Enter name to remove:");
                String nameToRemove = sc.nextLine();
                break;
            default:
                System.out.print ("Invalid");
        }
    }
}
```

3) Write a Java program that demonstrates the use of a Priority Queue to store employee names based on their priorities. Include functionality to:

- 1) Add employee names with different priorities.
- 2) Remove employee with highest priority.
- 3) Display the queue.

Input:-

- 1) Add Employee

- 2) Remove Employee with highest priority
- 3) Display employee Queue

- 4) Exit

Enter your choice: 1

Enter employee name to Add: Alice.

Output:- Employee added successfully.

Pseudo code:-

START

Initialize Priority Queue

Loop until user chooses to exit:

DISPLAY "1. Add Employee"

DISPLAY "2. Remove Employee"

DISPLAY "3. Display Employee Queue"

DISPLAY "4. Exit"

PROMPT User for choice

READ userChoice

IF userChoice is 1 THEN

PROMPT user to enter employee name

READ employeeName

ADD employeeName to employeeQueue

DISPLAY "Employee added successfully";

ELSE

DISPLAY "Queue is Empty"

IF employeeQueue is not empty THEN

DISPLAY "Employee Queue:"

ELSE

DISPLAY "Invalid"

END.

Program:-

```
import java.util.PriorityQueue;
```

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

```
public class EmployeePriorityQueue{
```

```
    public static void main(String[] args){
```

```
        PriorityQueue<String> employeeQueue = new
```

```
        PriorityQueue<>();
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        while(true){
```

```
            System.out.println("1. -- Employee priority
```

```
            System.out.println("2. Remove employee");
```

```
            System.out.println("3. Exit");
```

S.o.p ("3. Display Employee Queue")

S.o.p ("4. Exit")

S.o.p ("Enter your choice :");

switch (choice) {

case 1:

System.out.print("Enter employee name to add : ");

String employeeName = sc.nextLine();

break;

case 2:

S.o.p ("Enter employee name to remove : ");

String removedEmployee = employeeQueue.poll();

break;

case 4:

S.o.p ("Exit ");

sc.close();

return;

default:

System.out.println("Invalid ");

}

4) Create a Hashmap that stores student IDs and their names. Your program should:

- 1) Add key-value pairs to the Hashmap.
- 2) Search for a name using a student ID.
- 3) Remove a student using their ID.
- 4) Display all student entries.

Input:-

- 1) Add Student
- 2) Search for a Student by ID
- 3) Remove a Student by ID
- 4) Display All Students
- 5). Exit

Enter your choice: 1

Enter Student ID: 101

Enter Student name: Alice.

Output:- Student added successfully.

Pseudocode:-

START

Initialize Hashmap

LOOP until user chooses to exit:

DISPLAY "1. Add Student"

DISPLAY "2. Search Student"

DISPLAY "3. Remove Student"

DISPLAY "4. Display All Students"

DISPLAY "5. Exit".

PROMPT user for choice

READ userchoice

IF userchoice is 1 THEN

PROMPT user to enter student ID

READ idToAdd

PROMPT user to enter student name

READ nameToAdd

ADD idToAdd and nameToAdd to Student Map

DISPLAY "Student added successfully".

ELSE IF userchoice is 2 THEN

PROMPT user to enter student ID to search

READ idToSearch

IF StudentMap Contains idToSearch THEN

DISPLAY "Student Name: " + StudentMap.  
get (idToSearch)

ELSE

DISPLAY "Student ID not found".

ELSE IF userchoice is 3 THEN

PROMPT user to enter student ID to search

READ idToSearch

ELSE

DISPLAY "Student ID not found".

CLOSE any resources if necessary

END.

## Program:-

```
import java.util.HashMap;
import java.util.Scanner;
import java.util.Map;

public class StudentHashMap {
    public static void main(String[] args) {
        HashMap<Integer, String> studentMap =
            new HashMap<>();
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("In Student Management");
            System.out.println("1. Add Student");
            System.out.println("2. Search Student");
            System.out.println("3. Remove Student");
            System.out.println("4. Display students");
            System.out.println("Exit");
            System.out.print("Enter your choice:");
            switch (choice) {
                case 1:
                    System.out.print("Enter student ID:");
                    int idToAdd = sc.nextInt();
                    break;
                case 2:
                    System.out.print("Enter student ID :");
```

```
int idToSearch = sc.nextInt();
break;
```

Case 3:

```
S.o.p("Enter Student ID to remove!");
int idToRemove = sc.nextInt();
break;
```

Case 4:

```
if(StudentMap.isEmpty()){
    S.o.p("No Students to display");
} else {
    S.o.p("Student List:");
    for(Map.Entry<Integer, String> student
        Map.entrySet()) {
        S.o.p("ID: " + entry.getKey());
    }
}
```

default:

```
S.o.p("Invalid Choice");
}
}
}
}
```