SHRI VILEPARLE KELAVANI MANDAL'S

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Object Oriented Programming using Java Laboratory (DJS23FLES201)
Academic Year 2023-24

EXPERIMENT NO. 4

AIM / OBJECTIVE:

To implement collections (Array List/ Vectors)

DESCRIPTION OF EXPERIMENT:

Hierarchy of Collection Framework mentioning the different classes, interfaces

Benefits of collections

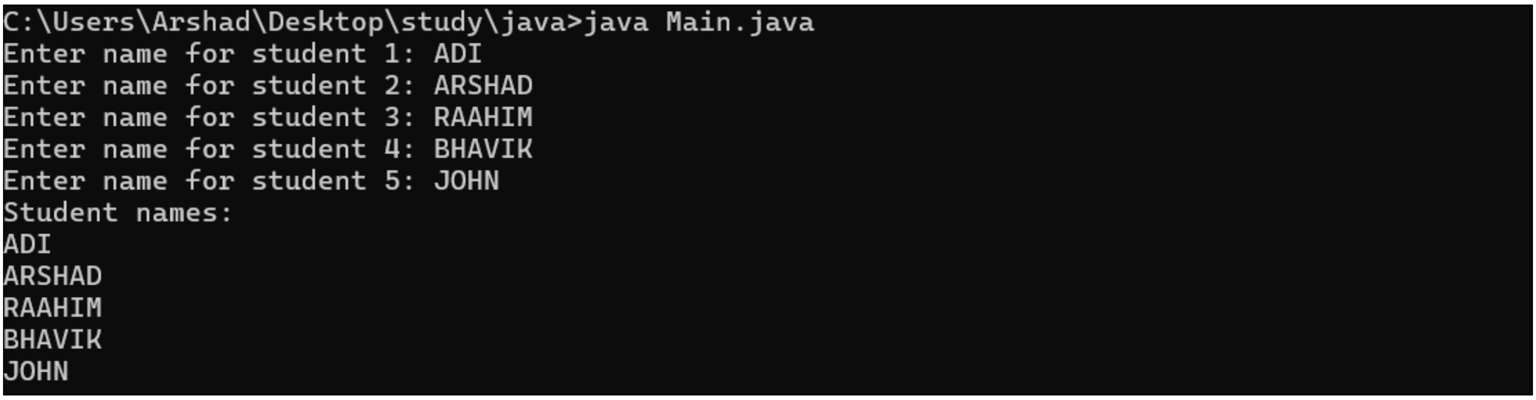
To implement collections (Array List/ Vectors)

a. WAP to accept students name from command line and store them in vector.

Code-

```
import java.util.Scanner;

import java.util.Vector;



public class StudentNameCollector {

    public static void main(String[] args) {

        Vector<String> studentNames = new Vector<>();



        Scanner scanner = new Scanner(System.in);



        // Accept names for 5 students
        for (int i = 0; i < 5; i++) {

            System.out.print("Enter name for student " + (i + 1) + ": ");

            String name = scanner.nextLine();

            studentNames.add(name);

        }



        // Display the collected student names

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

        for (String name : studentNames) {

            System.out.println(name);

        }

    }

}
```

Output -



```
C:\Users\Arshad\Desktop\study\java>java Main.java
Enter name for student 1: ADI
Enter name for student 2: ARSHAD
Enter name for student 3: RAAHIM
Enter name for student 4: BHAVIK
Enter name for student 5: JOHN
Student names:
ADI
ARSHAD
RAAHIM
BHAVIK
JOHN
```

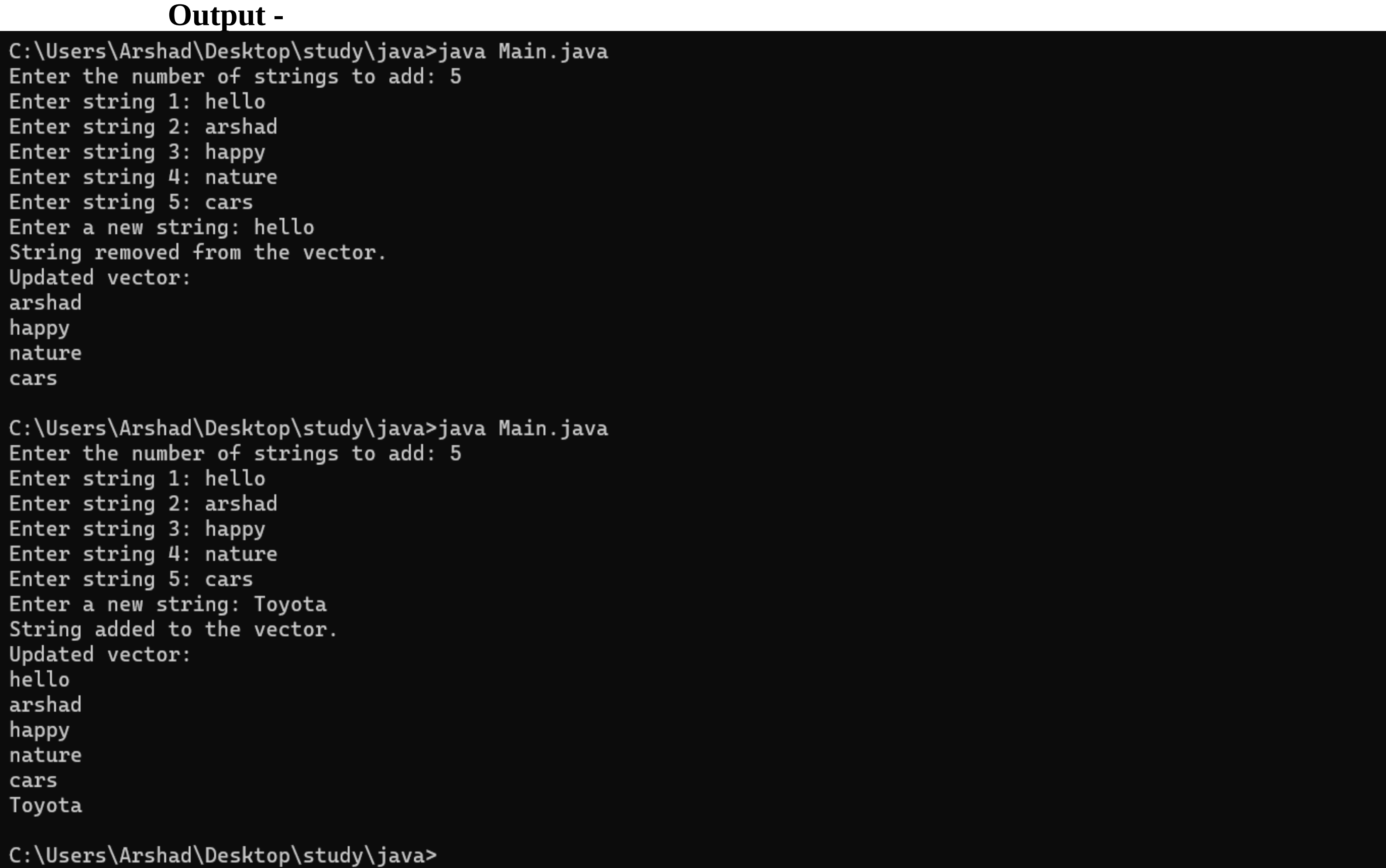
b. WAP to add n strings in a vector array. Input new string and check if it is present in the vector. If present delete it else add to the vector

Code-

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

public class StringVector {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Vector<String> strings = new Vector<>();

        // Input n strings
        System.out.print("Enter the number of strings to add: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume newline character

        for (int i = 0; i < n; i++) {
            System.out.print("Enter string " + (i + 1) + ": ");
            String inputString = scanner.nextLine();
    strings.add(inputString);

}



// Input new string

System.out.print("Enter a new string: ");

String newString = scanner.nextLine();



// Check if the new string is present in the vector

boolean found = false;

for (int i = 0; i < strings.size(); i++) {

    if (strings.get(i).equals(newString)) {

        strings.remove(i);

        found = true;

        break;

    }

}



// If not present, add it to the vector

if (!found) {

    strings.add(newString);

    System.out.println("String added to the vector.");
```

```
        } else {
            System.out.println("String removed from the vector.");
        }


        // Display the updated vector
        System.out.println("Updated vector:");
        for (String str : strings) {
            System.out.println(str);
        }
    }
}
```

Output -



```
C:\Users\Arshad\Desktop\study\java>java Main.java
Enter the number of strings to add: 5
Enter string 1: hello
Enter string 2: arshad
Enter string 3: happy
Enter string 4: nature
Enter string 5: cars
Enter a new string: hello
String removed from the vector.
Updated vector:
arshad
happy
nature
cars

C:\Users\Arshad\Desktop\study\java>java Main.java
Enter the number of strings to add: 5
Enter string 1: hello
Enter string 2: arshad
Enter string 3: happy
Enter string 4: nature
Enter string 5: cars
Enter a new string: Toyota
String added to the vector.
Updated vector:
hello
arshad
happy
nature
cars
Toyota

C:\Users\Arshad\Desktop\study\java>
```

CONCLUSION

We used the Vector class to store a dynamic collection of elements. Elements can be added to a vector using the add() method.

We accessed elements in the vector using the get() method, specifying the index of the element we want to retrieve.

We used loops (e.g., for loop) to iterate over the elements of the vector.

Elements can be removed from a vector using the remove() method, specifying either the index of the element or the element itself.

We searched for elements in the vector by iterating over its elements and comparing them with the target element.

Vectors automatically adjust their size as elements are added or removed, providing flexibility in managing collections of data.

These programmes demonstrate the versatility and usefulness of vectors in Java for managing collections of elements, particularly when the size of the collection may change over time.