Rajalakshmi Engineering College

Name: Charan Sundar V
Email: 241001036@rajalakshmi.edu.in
Roll no: 2116241001036
Phone: 7338594456
Branch: REC
Department: IT - Section 1
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 7_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 12

Section 1 : MCQ

1. Which of the following statements about Java interfaces is true?

Answer

A class can implement multiple interfaces.

Status : Correct ***Marks : 1/1***

2. How do you call a static method from an interface MyInterface?

Answer

MyInterface.staticMethod();

Status : Correct ***Marks : 1/1***

3. What is the output of the following code?

```
interface A {
    default void show() {
        System.out.println("A's Default Method");
    }
}

interface B {
    default void show() {
        System.out.println("B's Default Method");
    }
}

class C implements A, B {
    public void show() {
        A.super.show();
    }
}

public class Main {
    public static void main(String[] args) {
        C obj = new C();
        obj.show();
    }
}
```

Answer

Compilation Error

Status : Wrong ***Marks : 0/1***

4. Which of the following is the correct way to declare an interface in Java?

Answer

```
interface Vehicle {    void start();}
```

Status : Correct ***Marks : 1/1***

5. If a class implements two interfaces that have the same default method, what must the class do?

Answer

The class must override the method to resolve ambiguity.

Status : Correct ***Marks : 1/1***

6. Can a Java interface contain both default and static methods?

Answer

Yes, an interface can have both default and static methods.

Status : Correct ***Marks : 1/1***

7. Consider a class implementing an interface and extending a class, both having a method with the same name. Which method gets called?

Answer

The method from the superclass

Status : Correct ***Marks : 1/1***

8. Which of the following statements is true regarding default methods in Java interfaces?

Answer

A default method can be overridden in a class implementing the interface.

Status : Correct ***Marks : 1/1***

9. What is the output of the following code?

```
interface A {
    static void display() {
        System.out.println("Static method in A");
    }
```

```
}

class B implements A {
    static void display() {
        System.out.println("Static method in B");
    }
}

public class Main {
    public static void main(String[] args) {
        B.display();
    }
}
```

Answer

Static method in A

Status : Wrong ***Marks : 0/1***

10. What is the primary purpose of static methods in Java interfaces?

Answer

They allow an interface to provide helper methods without requiring an implementing class.

Status : Correct ***Marks : 1/1***

11. How can a class explicitly call a default method from an interface if there is a naming conflict?

Answer

Using InterfaceName.super.methodName();

Status : Correct ***Marks : 1/1***

12. What is the output of the following code?

```
interface A {
```

```
    default void show() {
        System.out.println("A's Default Method");
    }
}

class B {
    public void show() {
        System.out.println("B's Method");
    }
}

class C extends B implements A {
}

public class Main {
    public static void main(String[] args) {
        C obj = new C();
        obj.show();
    }
}
```

Answer

B's Method

Status : Correct ***Marks : 1/1***

13. What happens when an implementing class does not override a default method from an interface?

Answer

The default method's implementation from the interface will be used.

Status : Correct ***Marks : 1/1***

14. What is the output of the following code?

```
interface X {
    default void show() {
```

```
        System.out.println("X's Default Method");
    }
}

interface Y {
    default void show() {
        System.out.println("Y's Default Method");
    }
}

class Z implements X, Y {
    public void show() {
        System.out.println("Z's Method");
    }
}

public class Main {
    public static void main(String[] args) {
        Z obj = new Z();
        obj.show();
    }
}
```

Answer

Z's Method

Status : Correct ***Marks : 1/1***

15. What is the output of the following code?

```
interface MathOperations {
    static int square(int x) {
        return x * x;
    }
}

public class Main {
    public static void main(String[] args) {
        System.out.println(MathOperations.square(5));
```

```
    }
}
```

Answer

Compilation error

Status : Wrong ***Marks : 0/1***

2116241001036

Rajalakshmi Engineering College

Name: Charan Sundar V
Email: 241001036@rajalakshmi.edu.in
Roll no: 2116241001036
Phone: 7338594456
Branch: REC
Department: IT - Section 1
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bobby is tasked with processing a sequence of numbers from a monitoring system. He needs to extract a strictly increasing subsequence using an ArrayList. The program should dynamically add numbers to the ArrayList only if they are greater than the last number currently stored in the list. Bobby aims to efficiently utilize the dynamic resizing and indexing features of the ArrayList to solve this problem.

Help Bobby implement this solution.

Input Format

The first line of input consists of an integer N, representing the number of elements.

The second line consists of N space-separated integers, representing the elements.

Output Format

The output prints the list of integers in increasing sequence, ignoring out-of-order elements.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7
3 5 9 1 11 7 13
Output: [3, 5, 9, 11, 13]

Answer

```
// You are using Java
import java.util.ArrayList;
import java.util.Scanner;

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

      int N = sc.nextInt();
      int[] elements = new int[N];
      for (int i = 0; i < N; i++) {
         elements[i] = sc.nextInt();
      }

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

      for (int num : elements) {

         if (increasingList.isEmpty() || num >
increasingList.get(increasingList.size() - 1)) {
            increasingList.add(num);
         }
      }
```

```
        System.out.println(increasingList);
        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

2116241001036

Rajalakshmi Engineering College

Name: Charan Sundar V
Email: 241001036@rajalakshmi.edu.in
Roll no: 2116241001036
Phone: 7338594456
Branch: REC
Department: IT - Section 1
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Vikram loves listening to music and wants to create a simple playlist manager using Java Collections. The playlist supports the following operations:

"ADD <song>" Adds the song to the end of the playlist."REMOVE <song>" Removes the first occurrence of the song from the playlist. If the song is not found, do nothing."SHOW" Displays all songs in the playlist in order. If the playlist is empty, print "EMPTY"."NEXT" Moves to the next song in the playlist and prints its name. If the playlist is empty, print "EMPTY".

The playlist maintains a "current song" position that starts at the first song when it's added. The NEXT command moves to the next song and prints it, wrapping around to the first song after reaching the last song. When removing songs, the current position adjusts accordingly to maintain

proper navigation.

Help Vikram implement this playlist manager.

Input Format

The first line of the input consists of an integer n, the number of operations.

The next n lines, each containing a command:

- "ADD <song>"
- "REMOVE <song>"
- "SHOW"
- "NEXT"

Output Format

For each "SHOW" command, print the songs in order, separated by spaces.

For each "NEXT" command, print the next song in the playlist.

If no song exists, print "EMPTY".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7
ADD song1
ADD song2
SHOW
NEXT
REMOVE song2
SHOW
NEXT

Output: song1 song2
song2
song1
song1

Answer

2116241001036

```
// You are using Java
import java.util.LinkedList;
import java.util.Scanner;

class PlaylistManager {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        sc.nextLine();

        LinkedList<String> playlist = new LinkedList<>();
        int currentIndex = -1;

        for (int i = 0; i < n; i++) {
            String line = sc.nextLine();
            String[] parts = line.split(" ", 2);
            String command = parts[0];

            switch (command) {
                case "ADD":
                    String songToAdd = parts[1];
                    playlist.add(songToAdd);
                    if (playlist.size() == 1) {
                        currentIndex = 0;
                    }
                    break;

                case "REMOVE":
                    String songToRemove = parts[1];
                    int removeIndex = playlist.indexOf(songToRemove);
                    if (removeIndex != -1) {
                        playlist.remove(removeIndex);
                        if (playlist.isEmpty()) {
                            currentIndex = -1;
                        } else {
                            if (removeIndex < currentIndex) {
                                currentIndex--;
                            } else if (removeIndex == currentIndex) {
                                if (currentIndex >= playlist.size()) {
                                    currentIndex = 0;
                                }
                            }
```

2116241001036

```
                }
            }
            break;

        case "SHOW":
            if (playlist.isEmpty()) {
                System.out.print("EMPTY ");
            } else {
                for (int j = 0; j < playlist.size(); j++) {
                    System.out.print(playlist.get(j));
                    if (j != playlist.size() - 1) System.out.print(" ");
                }
                System.out.print(" ");
            }
            break;

        case "NEXT":
            if (playlist.isEmpty()) {
                System.out.print("EMPTY ");
            } else {
                currentIndex = (currentIndex + 1) % playlist.size();
                System.out.print(playlist.get(currentIndex) + " ");
            }
            break;
    }
  }
  sc.close();
 }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: Charan Sundar V
Email: 241001036@rajalakshmi.edu.in
Roll no: 2116241001036
Phone: 7338594456
Branch: REC
Department: IT - Section 1
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Assist Pranitha in developing a program that takes an integer N as input, representing the number of names to be read. Then read N names and store them in an ArrayList. Finally, input a search string and output the frequency of that string in the list of names.

Note: Some parts of the code are provided as snippets, and you need to complete the remaining sections by writing the necessary code.

Input Format

The first line of input consists of an integer N, representing the number of names to be read.

The following N lines consist of N names, as a string.

The last line consists of a string, representing the name to be searched.

Output Format

The output prints a single integer, representing the frequency of the specified name in the given list.

If the specified name is not found, print 0.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
Alice
Bob
Ankit
Alice
Pranitha
Alice

Output: 2

Answer

```
// You are using Java
import java.util.ArrayList;
import java.util.Scanner;

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


        int N = sc.nextInt();
        sc.nextLine();

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


        for (int i = 0; i < N; i++) {
            String name = sc.nextLine();
            names.add(name);
```

```
        }

        String searchName = sc.nextLine();

        int frequency = 0;
        for (String name : names) {
            if (name.equals(searchName)) {
                frequency++;
            }
        }

        System.out.println(frequency);

        sc.close();
    }
}
```

Status : Correct **Marks : 10/10**

2116241001036