System Requirements Specification Index

For

For Each Loop

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

USE CASE DESCRIPTION
System Requirements Specification

1 Project Abstract

This project assesses knowledge of Java looping constructs, specifically the **for-each loop**.
The tasks involve performing operations such as iterating over arrays and strings, summing elements, finding maximum values, and counting occurrences using for-each loops.

2 Assessment Tasks

Task 1: Print Array Elements Using For-Each Loop:

- Prompt the user: `"Enter the number of elements in the array:"`.
- Accept the input and store it in an integer variable `n`.
- Declare an integer array `array` of size `n`.
- Prompt the user: `"Enter the elements of the array:"`.
- Use a standard `for` loop from `0` to `n-1` to accept and store the elements in the array.
- Print a message: `"Array elements:"`.
- Use a **for-each loop** to iterate through each `element` in the `array`:
  - ➔ Print each `element` on a separate line.

Expected Output:

Enter the number of elements in the array:
5
Enter the elements of the array:
1 2 3 4 5
Array elements:
1
2
3
4
5

Task 2: Calculate Sum of Elements in an Array Using For-Each Loop:

- Declare and initialize an integer variable `sum` with `0`.
- Use a **for-each loop** to iterate through each `element` in the `array`:
  - ➔ Add each `element` to `sum`.
- After the loop completes, print: `"Sum of elements: <sum>"`.

Expected Output:

Sum of elements: 15

Task 3: Print Characters of a String Using For-Each Loop:

- Prompt the user: `"Enter a string:"`.
- Accept the input and store it in a `String` variable `str`.
- Print a message: `"Characters of the string:"`.
- Convert the string to a character array using `toCharArray()`.
- Use a **for-each loop** to iterate through each `char ch` in the character array:
  - ➔ Print each character on a separate line.

Expected Output:

```
Enter a string: hello
Characters of the string:
h
e
l
l
o
```

Task 4: Find Maximum Element in an Array Using For-Each Loop:

- Declare and initialize an integer variable `max` with the first element of the `array`.
- Use a **for-each loop** to iterate through each `element` in the `array`:
  - ➔ If `element` is greater than `max`, update `max` with `element`.
- After the loop, print: `"Maximum element: <max>"`.

Expected Output:

```
Maximum element: 5
```

Task 5: Count Occurrences of a Specific Element in an Array Using For-Each Loop:

- Prompt the user: `"Enter the element to search for:"`.
- Accept the input and store it in an integer variable `target`.
- Declare and initialize an integer variable `count` with `0`.
- Use a **for-each loop** to iterate through each `element` in the `array`:
  - ➔ If `element` is equal to `target`, increment `count` by `1`.
- After the loop, print: `"Occurrences of <target>: <count>"`.

Expected Output:

```
Enter the element to search for: 3
Occurrences of 3: 1
```

Note: The actual output values will vary depending on the user inputs.

3 Template Code Structure

3.1 Package: com.yaksha.assignment.ForEachLoopAssignment

Resources

Class/Interface	Description	Status
ForEachLoopAssignment (class)	● Main class demonstrating iterative operations using **for-each loops**. ● Includes examples of: - Printing array elements using **for-each loop**. - Calculating sum of elements in an array. - Printing characters of a string. - Finding maximum element in an array. - Counting occurrences of a specific element.	Need to be implemented.

4 Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) Terminal New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**

5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **To run your project use command:**
   mvn compile exec:java
   -Dexec.mainClass="com.yaksha.assignment.ForEachLoopAssignment"
7. **To test your project test cases, use the command**
   mvn test
8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**