System Requirements Specification Index

For

Basic Array using For each

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

1 Project Abstract

This project assesses knowledge of Java arrays and the use of the **for-each loop** to perform operations.
The tasks include iterating over arrays to print elements, calculate length, find maximum values, skip specific elements, and compute the sum of array elements using **for-each loops**.

2 Assessment Tasks

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

- Print the message:
  `"Task 1: Print Elements of an Array"`.
- Declare and initialize an integer array `arr1` with elements `{1, 2, 3, 4, 5}`.
- Use a **for-each loop** to iterate through each element `num` in `arr1`:
  ➔ Print each `num` on a separate line.
- This will print all elements of `arr1`.

Expected Output:

```
Task 1: Print Elements of an Array
1
2
3
4
5
```

Task 2: Find the Length of an Array:

- Print the message:
  `"Task 2: Find the Length of an Array"`.
- Declare and initialize an integer array `arr2` with elements `{1, 2, 3, 4, 5}`.
- Use `arr2.length` to find the length of the array and print:
  `"Length of the array: <length>"`.
- This will display the total number of elements in `arr2`.

Expected Output:

```
Task 2: Find the Length of an Array
Length of the array: 5
```

Task 3: Find the Maximum Value in an Array Using For-Each Loop:

- Print the message:
  `"Task 3: Find the Maximum Value in an Array"`.

- Declare and initialize an integer array `arr3` with elements `{1, 2, 3, 4, 5, 9}`.
- Declare an integer variable `max` and initialize it with the first element of `arr3`.
- Use a **for-each loop** to iterate through each element `num` in `arr3`:
  - ➔ In each iteration, check if `num` is greater than `max`:
    - If true, update `max` with `num`.
- After the loop, print:
  `"Maximum Value: <max>"`.
- This will print the maximum element in the array.

Expected Output:

```
Task 3: Find the Maximum Value in an Array
Maximum Value: 9
```

Task 4: Skip a Specific Value in an Array Using Continue Keyword:

- Print the message:
  `"Task 4: Skip a Specific Value in an Array"`.
- Declare and initialize an integer array `arr4` with elements `{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}`.
- Use a **for-each loop** to iterate through each element `num` in `arr4`:
  - ➔ Inside the loop, check if `num` equals `7`:
    - If true, use `continue` to skip the current iteration.
  - ➔ Otherwise, print the value of `num`.
- This will print all elements except `7`.

Expected Output:

```
Task 4: Skip a Specific Value in an Array
1
2
3
4
5
6
8
9
10
```

Task 5: Calculate Sum of Elements in an Array Using For-Each Loop:

- Print the message:
  `"Task 5: Sum of Elements in an Array"`.
- Declare and initialize an integer array `arr5` with elements `{1, 2, 3, 4, 5}`.
- Declare an integer variable `sum` and initialize it to `0`.
- Use a **for-each loop** to iterate through each element `num` in `arr5`:
  - ➔ Add each `num` to `sum`.
- After the loop completes, print:

`"Sum of the array elements: <sum>"`.

- This will print the total sum of the array elements.

Expected Output:

Sum of the array elements: 15

3 Template Code Structure

3.1 Package: com.yaksha.assignment.ArrayUsingForEachAssignment

Resources

Class/Interface	**Description**	**Status**
ArrayUsingForEachAssignment (class)	● Main class demonstrating basic operations on **arrays using for-each loops**. ● Includes examples of: - Printing elements of an array using **for-each loop**. - Finding the length of an array. - Finding the maximum value in an array. - Skipping specific values. - Calculating sum of elements.	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.ArrayUsingForEachAssignment"
   ```

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.**