
System Requirements Specification Index

For

Nested For Loop

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1	Project Abstract	3
2	Assessment Tasks	3
3	Template Code Structure	5
3.1	Package: com.yaksha.assignment.NestedForLoopAssignment	5
4	Execution Steps to Follow	6

USE CASE DESCRIPTION

System Requirements Specification

1 PROJECT ABSTRACT

This project assesses knowledge of Java looping constructs, specifically **nested for loops**. The tasks involve performing nested iterations to solve problems such as printing patterns, generating multiplication tables, identifying prime numbers, and constructing Floyd's triangle.

2 ASSESSMENT TASKS

Task 1: Print a Right-Angled Triangle Pattern:

- Prompt the user: "Enter the number of rows for the triangle:".
- Accept the input and store it in integer variable `rows`.
- Use an outer `for` loop with `i` starting from 1 to `rows`:
 - Inside the outer loop, use an inner `for` loop with `j` starting from 1 to `i`:
 - In each iteration, print "*" without a new line.
- After the inner loop completes, print a new line using `System.out.println()`.
- The pattern will form a right-angled triangle.

Expected Output:

```
Enter the number of rows for the triangle: 4
*
**
***
****
```

Task 2: Print Multiplication Table Grid Using Nested For Loops:

- Print a message: "Multiplication table grid from 1 to 5:".
- Use an outer `for` loop with `i` starting from 1 to 5:
 - Inside the outer loop, use an inner `for` loop with `j` starting from 1 to 5:
 - In each iteration, calculate `i * j` and print the result in the format:
 - "<i> x <j> = <product>" followed by a tab space.
 - After the inner loop completes, print a new line using `System.out.println()`.

Expected Output:

```
Multiplication table grid from 1 to 5:
1 x 1 = 1    1 x 2 = 2    1 x 3 = 3    1 x 4 = 4    1 x 5 = 5
2 x 1 = 2    2 x 2 = 4    2 x 3 = 6    2 x 4 = 8    2 x 5 = 10
3 x 1 = 3    3 x 2 = 6    3 x 3 = 9    3 x 4 = 12   3 x 5 = 15
4 x 1 = 4    4 x 2 = 8    4 x 3 = 12   4 x 4 = 16   4 x 5 = 20
5 x 1 = 5    5 x 2 = 10   5 x 3 = 15   5 x 4 = 20   5 x 5 = 25
```

Task 3: Find Prime Numbers Within a Range Using Nested For Loops:

- Prompt the user: "Enter the lower limit:".
- Accept the input and store it in an integer variable `lowerLimit`.
- Prompt the user: "Enter the upper limit:".
- Accept the input and store it in an integer variable `upperLimit`.
- Print a message: "Prime numbers between <lowerLimit> and <upperLimit>:".
- Use an outer `for` loop with `num` starting from `lowerLimit` to `upperLimit`:
 - Declare a boolean variable `isPrime` and initialize it to `true`.
 - Use an inner `for` loop with `i` starting from `2` to `Math.sqrt(num)`:
 - If `num` is divisible by `i`, set `isPrime` to `false` and break the loop.
 - After the inner loop, check if `isPrime` is `true` and `num > 1`:
 - If true, print `num` followed by a space.
- After the loop completes, print a new line.

Expected Output:

```
Enter the lower limit: 10
Enter the upper limit: 20
Prime numbers between 10 and 20:
11 13 17 19
```

Task 4: Print Floyd's Triangle Using Nested For Loops:

- Prompt the user: "Enter the number of rows for Floyd's Triangle:".
- Accept the input and store it in an integer variable `n`.
- Declare and initialize an integer variable `number` with `1`.
- Use an outer `for` loop with `i` starting from `1` to `n`:
 - Inside the outer loop, use an inner `for` loop with `j` starting from `1` to `i`:
 - In each iteration, print the value of the `number` followed by a space and increment the `number` by `1`.
 - After the inner loop completes, print a new line using `System.out.println()`.

Expected Output:

```
Enter the number of rows for Floyd's Triangle: 4
1
2 3
4 5 6
7 8 9 10
```

Task 5: Create a Multiplication Table of Any Number Using For Loop:

- Prompt the user: "Enter a number for the multiplication table:".

- Accept the input and store it in an integer variable `num`.
- Use a `for` loop with `i` starting from `1` to `10`:
 - In each iteration, calculate `num * i` and print the result in the format: "`<num> x <i> = <product>`".

Expected Output:

Enter a number for the multiplication table: 4

4 x 1 = 4

4 x 2 = 8

4 x 3 = 12

4 x 4 = 16

4 x 5 = 20

4 x 6 = 24

4 x 7 = 28

4 x 8 = 32

4 x 9 = 36

4 x 10 = 40

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

3 TEMPLATE CODE STRUCTURE

3.1 PACKAGE: `COM.YAKSHA.ASSIGNMENT.NESTEDFORLOOPASSIGNMENT`

Resources

Class/Interface	Description	Status
NestedForLoopAssignment (class)	<ul style="list-style-type: none"> • Main class demonstrating iterative operations using nested for loops. • Includes examples of: <ul style="list-style-type: none"> - Printing a right-angled triangle pattern. - Generating a multiplication table grid. - Finding prime numbers within a range. - Constructing Floyd's triangle. - Printing the multiplication table of a number. 	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.NestedForLoopAssignment"
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.