
System Requirements Specification Index

For

Break – Continue

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.BreakContinueAssignment	5
4	Execution Steps to Follow	6

USE CASE DESCRIPTION

System Requirements Specification

1 PROJECT ABSTRACT

This project assesses knowledge of Java control flow statements, specifically the usage of **break** and **continue** keywords.

The tasks involve controlling loop execution by breaking out of loops based on conditions and skipping specific iterations using continue statements.

2 ASSESSMENT TASKS

Task 1: Print Numbers Until a Specific Condition is Met Using Break Keyword:

- Print the message:
"Task 1: Print Numbers until a Specific Condition is Met (Break)".
- Use a **for** loop with **i** starting from 1 to 10:
 - Inside the loop, use an **if** condition to check if **i > 5**.
 - If true, use **break** to terminate the loop.
 - Otherwise, print the value of **i**.
- This will print numbers from 1 to 5 and stop when **i** exceeds 5.

Expected Output:

Task 1: Print Numbers until a Specific Condition is Met (Break)

1
2
3
4
5

Task 2: Skip Even Numbers Using Continue Keyword:

- Print the message:
"Task 2: Skip Even Numbers (Continue)".
- Use a **for** loop with **i** starting from 1 to 10:
 - Inside the loop, use an **if** condition to check if **i** is even (**i % 2 == 0**).
 - If true, use **continue** to skip the current iteration.
 - Otherwise, print the value of **i**.
- This will print all **odd numbers** from 1 to 10 and skip the even numbers.

Expected Output:

Task 2: Skip Even Numbers (Continue)

1
3

5
7
9

Task 3: Find First Odd Number Greater Than a Given Value Using Break Keyword:

- Print the message:
"Task 3: Find First Odd Number Greater than a Given Value (Break)".
- Use a **for** loop with **i** starting from 1 to 20:
 - Inside the loop, check if **i** is greater than 15 and **i** is odd (**i % 2 != 0**).
 - If true, print **i** and use **break** to terminate the loop.
- This will print the first odd number greater than 15 (i.e., 17).

Expected Output:

Task 3: Find First Odd Number Greater than a Given Value (Break)
17

Task 4: Skip Specific Number Using Continue Keyword:

- Print the message:
"Task 4: Skip Specific Number (Continue)".
- Use a **for** loop with **i** starting from 1 to 10:
 - Inside the loop, use an **if** condition to check if **i == 7**.
 - If true, use **continue** to skip the current iteration.
 - Otherwise, print the value of **i**.
- This will print all numbers from 1 to 10, except 7.

Expected Output:

Task 4: Skip Specific Number (Continue)
1
2
3
4
5
6
8
9
10

Task 5: Print Numbers and Skip Multiples of 3 Using Continue Keyword:

- Print the message:
"Task 5: Print Numbers and Skip Multiples of 3 (Continue)".
- Use a **for** loop with **i** starting from 1 to 20:
 - Inside the loop, use an **if** condition to check if **i** is divisible by 3 (**i % 3 == 0**).

- If true, use **continue** to skip the current iteration.
→ Otherwise, print the value of **i**.
- This will print numbers from **1** to **20**, skipping all multiples of **3**.

Expected Output:

Task 5: Print Numbers and Skip Multiples of 3 (Continue)

```
1
2
4
5
7
8
10
11
13
14
16
17
19
20
```

3 TEMPLATE CODE STRUCTURE

3.1 PACKAGE: COM.YAKSHA.ASSIGNMENT.BREAKCONTINUEASSIGNMENT

Resources

Class/Interface	Description	Status
BreakContinueAssignment (class)	<ul style="list-style-type: none"> • Main class demonstrating usage of break and continue keywords. • Includes examples of: <ul style="list-style-type: none"> - Breaking a loop when a specific condition is met. - Skipping even numbers. - Finding first odd number greater than a value and breaking loop. - Skipping specific numbers or multiples using continue. 	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.BreakContinueAssignment"
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.