

ASSIGNMENT-01(17-08-2023)

NAME: Santhosh S

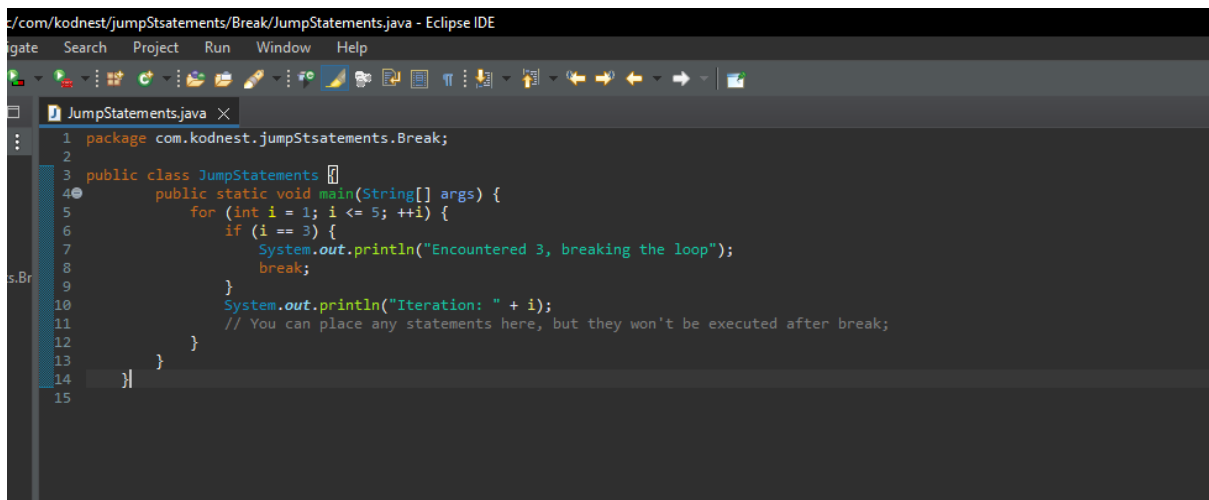
BATCH: July(A2)

Can We add any statements immediately below the break or continue ?

Break:

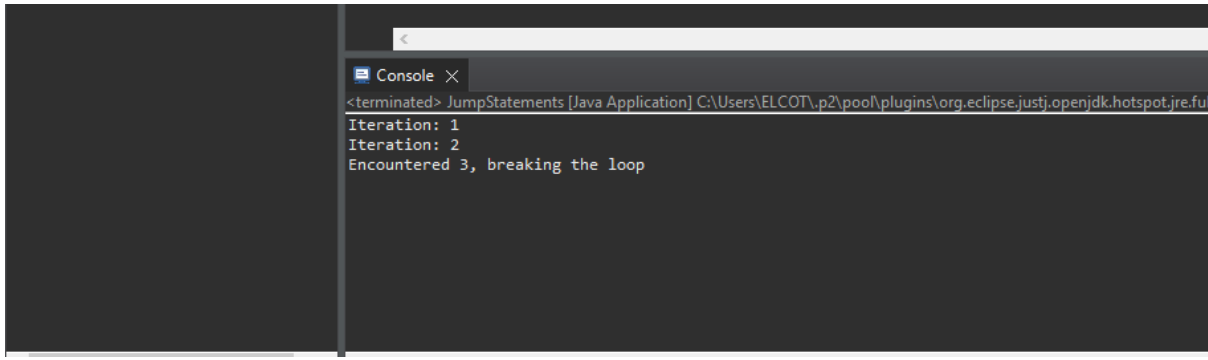
we can place any statements below the break; statement, but those statements will not be executed when the break; statement is encountered.

Example:

A screenshot of the Eclipse IDE interface. The title bar shows the file path: c:/com/kodnest/jumpStatements/Break/JumpStatements.java - Eclipse IDE. The menu bar includes File, Edit, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, running, and debugging. The editor window shows the following Java code:

```
1 package com.kodnest.jumpStatements.Break;
2
3 public class JumpStatements {
4     public static void main(String[] args) {
5         for (int i = 1; i <= 5; ++i) {
6             if (i == 3) {
7                 System.out.println("Encountered 3, breaking the loop");
8                 break;
9             }
10            System.out.println("Iteration: " + i);
11            // You can place any statements here, but they won't be executed after break;
12        }
13    }
14 }
15
```

Output:



```
<terminated> JumpStatements [Java Application] C:\Users\ELCOT\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre\bin\java.exe
Iteration: 1
Iteration: 2
Encountered 3, breaking the loop
```

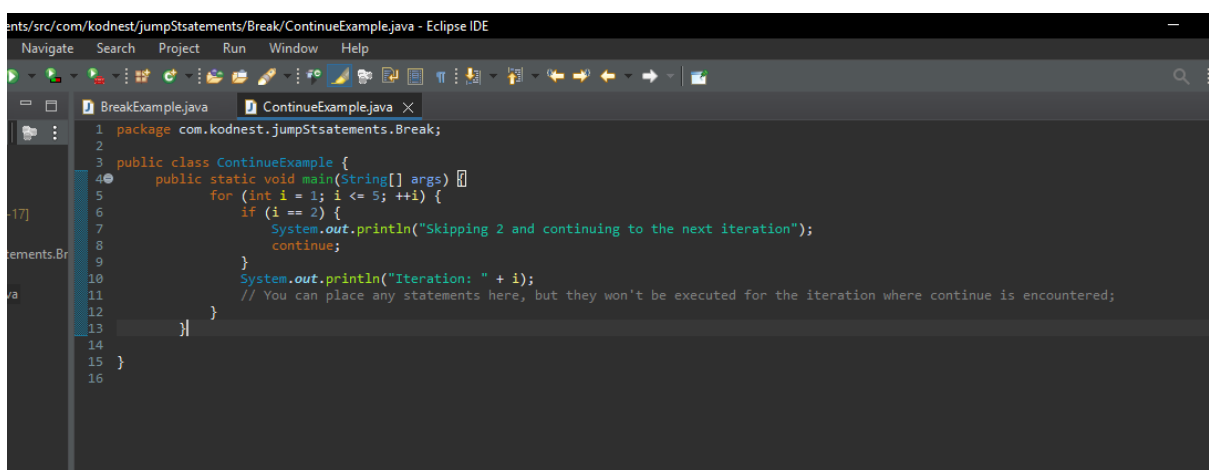
In this Example, you can place additional statements after the break; line, but they won't be executed due to the immediate termination of the loop.

It's important to note that these statements will never be reached when the break; statement is encountered

Continue:

We can place any statements below the continue; statement, but those statements will not be executed when the continue; statement is encountered.

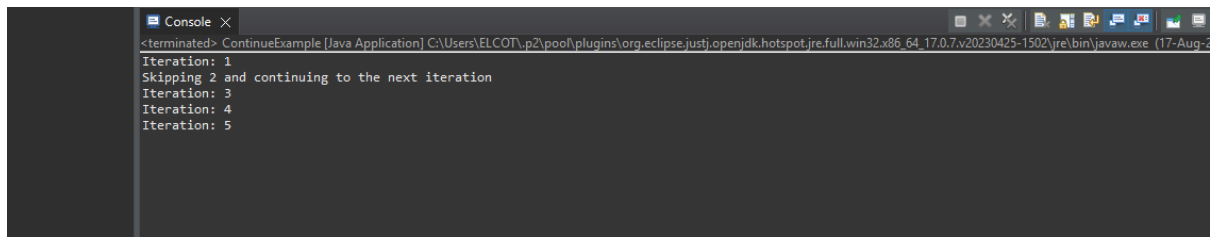
Example:



```
src/com/kodnest/jumpStatements/Break/ContinueExample.java - Eclipse IDE
package com.kodnest.jumpStatements.Break;

public class ContinueExample {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; ++i) {
            if (i == 2) {
                System.out.println("Skipping 2 and continuing to the next iteration");
                continue;
            }
            System.out.println("Iteration: " + i);
            // You can place any statements here, but they won't be executed for the iteration where continue is encountered;
        }
    }
}
```

Output:

A screenshot of a Java console window titled "Console X". The window shows the output of a Java application. The first line is "<terminated> ContinueExample [Java Application] C:\Users\ELCOT\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (17-Aug-2023)". The subsequent lines are "Iteration: 1", "Skipping 2 and continuing to the next iteration", "Iteration: 3", "Iteration: 4", and "Iteration: 5". The console has a dark background with light-colored text.

```
<terminated> ContinueExample [Java Application] C:\Users\ELCOT\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (17-Aug-2023)
Iteration: 1
Skipping 2 and continuing to the next iteration
Iteration: 3
Iteration: 4
Iteration: 5
```

In this Example, We can place additional statements after the continue; line, but they won't be executed for the current iteration due to the continue statement skipping to the next iteration.

These statements will not be reached when the continue; statement is encountered.