

NAME: JUNAID GIRKAR
SAP ID: 60004190057
DIVISION/BRANCH: SE - A3 - COMPS

AIM:

Program on branching, looping, labelled break and labelled continue.

THEORY:

LOOPING & BRANCHING

Java programming language provides following types of decision making or branching statements.

Java programming language provides following types of decision-making statements.

Sr No.	Statement & Description
	if statement
1	An if statement consists of a Boolean expression followed by one or more statements.
	if...else statement
2	An if statement can be followed by an optional else statement, which executes when the Boolean expression is false.
	nested if statement
3	You can use one if or else if statement inside another if or else if statement(s).
	switch statement
4	A switch statement allows a variable to be tested for equality against a list of values.

Java provides three branching statements break, continue and return.

Branching statements in Java are used to change the normal flow of execution based on some condition.

The return branching statement is used to explicitly return from a method.

break branching statement is used to break the loop and transfer control to the line immediate outside of loop.

continue branching statement is used to escape current execution and transfers control back to the start of the loop.

Java branching statements break and continue statements have two forms: the labeled form and the unlabeled form.

The LABELED BREAK Statement

Sometimes, it is a requirement of the program logic to exit from the entire nested control statement with just one statement rather than wait for it to complete the entire execution. This type of situation is particularly suitable for a labeled *break* statement. This statement enables one to break from the entire labeled block in one go. The program execution resumes from the first statement encountered after the enclosing labeled break statement.

When the labeled *break* statement is executed, it skips all the remaining operations from that point to the end of the enclosing labeled block and resumes execution thereafter. Therefore, the labeled break statement is a flexible way to define the usual length of the jump by using the unlabeled *break* statements.

The LABELED CONTINUE Statement

The labeled *continue* statement is similar to the unlabeled *continue* statement in the sense that both resume the iteration. The difference with the labeled continue statement is that it resumes operation from the target label defined in the code. As soon as the labeled continue is encountered, it skips the remaining statements from the statement's body and any number of enclosing loops and jumps to the next iteration of the enclosing labeled loop statements.

The *for* loop has been started with a label. When the *continue* statement is executed, it jumps to the target label and begins the iteration afresh. And, other statements after the *continue* operation are simply skipped.

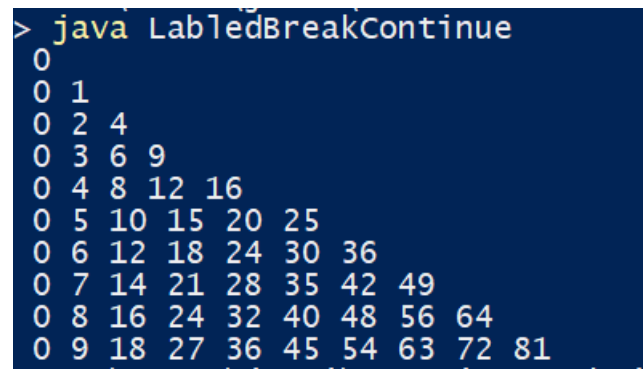
CONCLUSION:

The labeled break and continue statements are the only way to write statements similar to goto, since Java does not support goto statements. In this experiment, we have successfully executed labeled break and continue statements using branching and looping.

CODE:

```
public class Main
{
    public static void main(String[] args) {
        int breaklimit = 9;
        outer: for (int i = 0; ; i++) {
            for (int j = 0; j < 10; j++) {
                if (j > i) {
                    System.out.println();
                    continue outer;
                }
                System.out.print(" " + (i * j));
            }
            if(i==breaklimit){
                break outer;
            }
        }
        System.out.println();
    }
}
```

OUTPUT:



```
> java LabeledBreakContinue
0
0 1
0 2 4
0 3 6 9
0 4 8 12 16
0 5 10 15 20 25
0 6 12 18 24 30 36
0 7 14 21 28 35 42 49
0 8 16 24 32 40 48 56 64
0 9 18 27 36 45 54 63 72 81
```

```
C:\Windows\System32\cmd.exe
E:\Engineering\Semester - 3\College Stuff\Java\AP Sir Practicals\Experiment 2>javac LabelledBreakandContinue.java
E:\Engineering\Semester - 3\College Stuff\Java\AP Sir Practicals\Experiment 2>java LabelledBreakandContinue
[1 = 0, j = 0]
[1 = 0, j = 1]
[1 = 0, j = 2]
[1 = 2, j = 0]
[1 = 2, j = 1]
[1 = 2, j = 2]
[1 = 3, j = 0]
[1 = 3, j = 1]
[1 = 3, j = 2]
[1 = 0, j = 0]
[1 = 0, j = 1]
[1 = 0, j = 2]
E:\Engineering\Semester - 3\College Stuff\Java\AP Sir Practicals\Experiment 2>
```