

Java break and continue Statements

Java break and continue statements are used to manage program flow. We can use them in a loop to control loop iterations. These statements let us to control loop and switch statements by enabling us to either break out of the loop or jump to the next iteration by skipping the current loop iteration.

Break Statement

In Java, break is a statement that is used to break current execution flow of the program.

We can use break statement inside loop, switch case etc.

If break is used inside a loop then it will terminate the loop.

If break is used inside the innermost loop, then break will terminate the innermost loop only and execution will start from the outer loop.

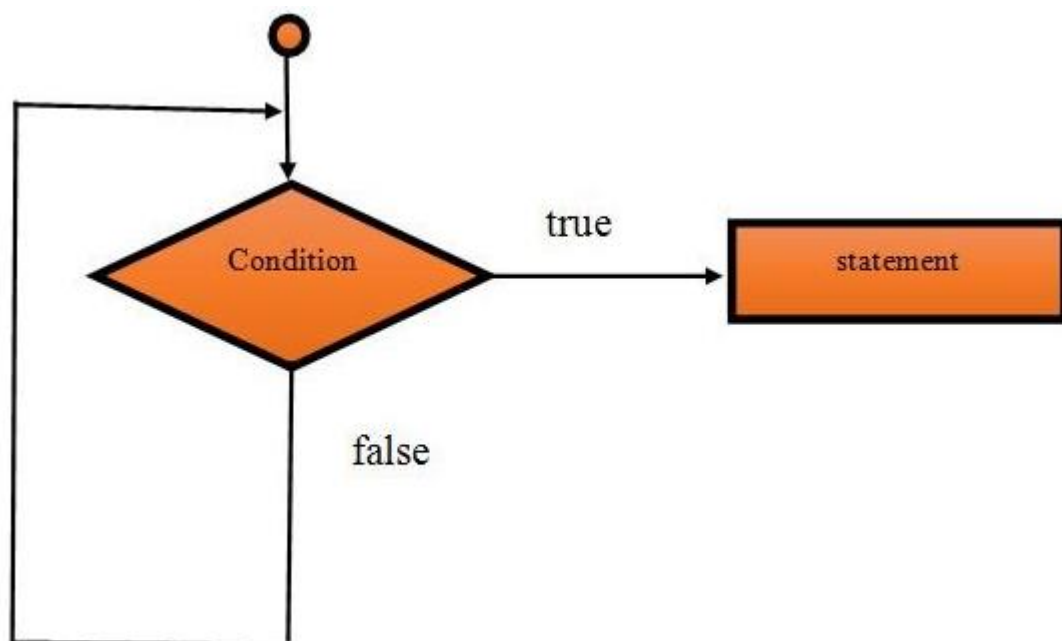
If break is used in switch case, then it will terminate the execution after the matched case. Use of break, we have covered in our switch case topic.

- Syntax:

jump-statement;

break;

- Data Flow Diagram of break statement



- Example:

In this example, we are using break inside the loop, the loop will terminate when the value is 8.

```
class BreakDemo1 {  
    public static void main(String[] args) {  
        for(int i=1;i<=10;i++){  
            if(i==8){  
                break;  
            }  
            System.out.println(i);  
        }  
    }  
}
```

- Example using break in do while loop

Loop can be any one whether it is for or while, break statement will do the same. Here, we are using break inside the do while loop.

```
class BreakDoWhileDemo1 {  
    public static void main(String[] args) {  
        int i=1;  
        do{  
            if(i==15)  
            {  
                i++;  
                break;  
            }  
            System.out.println(i);  
            i++;  
        }while(i<=20);  
    }  
}
```

- Example: Break in innermost loop

In this example, we are using break inside the innermost loop. But the loop breaks each time when j is equal to 2 and control goes to outer loop that starts from the next iteration.

```
class Demo{  
    public static void main(String[] args) {  
        for(int i=1;i<=2;i++){  
            for (int j = 0; j <=3; j++) {  
                if(j==2)  
                    break;  
                System.out.println(j);  
            }  
        }  
    }  
}
```

continue Statement

In Java, the continue statement is used to skip the current iteration of the loop. It jumps to the next iteration of the loop immediately. We can use continue statement with for loop, while loop and do-while loop as well.

jump-statement;

continue;

- Example:

In this example, we are using continue statement inside the for loop. See, it does not print 5 to the console because at fifth iteration continue statement skips the iteration that's why print statement does not execute.

```
class ContinueDemo1  
{  
    public static void main(String[] args)  
    {  
        for(int i=1;i<=10;i++)
```

```

        {
            if(i==5)
            {
                continue;
            }
            System.out.println(i);
        }
    }
}

```

- Example:

We can use label along with continue statement to set flow control. By using label, we can transfer control at specified location.

In this example, we are transferring control to outer loop by using label.

```

class ContinueDemo2 {
    public static void main(String[] args) {
        xy:
        for(int i=1;i<=5;i++){
            pq:
            for(int j=1;j<=5;j++){
                if(i==2&& j==2){
                    continue xy;
                }
                System.out.println(i+" "+j);
            }
        }
    }
}

```