



# Class Notes

*Published Date: November, 2022*

## PPS: UNIT-3

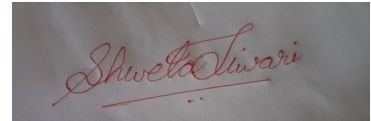
### Iteration and Loop

*FALL SEMESTER, YEAR (I/II sem, 1st yr)*

*FALL SESSION (2022-23)  
(PPS)*

*MS. SHWETA TIWARI*

*Published Date: November, 2022*



[shwetatiwario8@recabn.ac.in](mailto:shwetatiwario8@recabn.ac.in)  
[shwetatiwario8aug@gmail.com](mailto:shwetatiwario8aug@gmail.com)

## **TOPIC On : UNIT-3: Difference between Break and Continue Statement in C**

By SHWETA TIWARI  
**Under On: Iteration and Loop**

**PREPARED FOR**  
Engineering Students  
All Engineering College

**PREPARED BY**  
SHWETA TIWARI

# Difference between break and continue statement in C

**break statement:** This statement terminates the smallest enclosing loop (i.e., **while**, do-while, for loop, or switch statement).

**continue statement:** This statement skips the rest of the loop statement and starts the next iteration of the loop to take place.

---

Both the statements are of a similar type, and they allow a user to alter/ make changes to the flow of a program. But they are still different. The primary difference between break and continue statement in C is that the break statement leads to an immediate exit of the innermost switch or enclosing loop. On the other hand, the continue statement begins the next iteration of the while, enclosing for, or do loop. In the cases of while and do loops, the continue statement immediately takes control of their *test condition*. Similarly, it takes control of the loop's *increment* step in the case of a for a loop.

We can only apply the continue statement on the loops and not to any switch. When a continue is present inside a switch (further inside a loop), it causes the iteration of the next loop.

Practically, we can use a break in the switch if we want to exit after the execution of a particular case. The break also helps in the case of a loop whenever we wish to leave the loop after the occurrence of a certain condition. For example, if you prematurely reach the end of your data or reach an error condition. We use the continue statement only when we are willing to skip a statement (or more than one) in the body of a loop and transfer its overall control to its next iteration.

---

## Difference Between Break and Continue Statement in C

Parameters	Break Statement in C	Continue Statement in C
<b>Loop Construct</b>	This statement lets a user exit from an overall loop construct.	It does not let a user make an exit from an overall loop construct.
<b>Switch and Loop Statement</b>	One can easily use the break statement along with the switch statement. You can also use it within the for loop, do-while loop, and the while loop. It means that break can easily occur in both loop and switch.	You cannot use the continue statement with the switch statement. Still, you can use it within the for loop, do-while loop, and the while loop. It means that continue can only occur in the loop and not switch.
<b>Control</b>	The control exits immediately from a loop construct as soon as it encounters the break statement.	As soon as the control encounters the continue statement, it passes automatically from the very beginning of a loop statement.
<b>Function</b>	The break statement causes a loop or a switch to terminate a case at the very moment of its execution. It means that a switch or a loop would end abruptly as soon as they encounter a break.	The continue statement doesn't cause a loop termination- but leads it into its next iteration. It means that a loop will execute all of its iterations even if it encounters a continue statement. We use the continue statement to skip those statements that appear after the continue in a loop.
<b>Syntax</b>	You can denote it as: <b>break;</b>	You can denote it as: <b>continue;</b>