

Introduction to 'C' Language - Module 6

Conditional Statements and Loops in C

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

In C programming, break, goto, and continue statements are control flow mechanisms that alter the execution sequence of a program. Let's explore each of these statements in detail:

1. Break Statement

The break statement is used to exit a loop or switch statement prematurely, meaning it immediately terminates the loop or switch and transfers control to the statement that follows the loop or switch.

Usage in Loops:

- Example (in a loop):

```
c Copy code  
  
for (int i = 1; i <= 10; i++) {  
    if (i == 5) {  
        break; // Exit the loop when i equals 5  
    }  
    printf("%d\n", i);  
}
```

Output:

```
Copy code  
  
1  
2  
3  
4
```

```
default:  
    printf("Invalid day\n");  
}
```

Output:

```
mathematica Copy code  
  
Wednesday
```

2. Goto Statement

The goto statement provides a way to transfer control to another part of the program. It's generally considered bad practice to use goto as it can make code harder to follow and maintain. However, in some cases, it might be useful for jumping to a specific section of code, like breaking out of multiple nested loops.

Usage:

- Example:

```
c Copy code  
  
int i = 1;  
start:  
printf("%d\n", i);  
i++;  
if (i <= 5) {  
    goto start; // Jump back to the label 'start'  
}
```

Output:

```
Copy code  
  
1  
2  
3  
4  
5
```

3. Continue Statement

The continue statement skips the current iteration of a loop and moves control back to the beginning of the loop for the next iteration. It does not terminate the loop but skips the remaining code inside the loop for the current iteration.

 CBSE

 ICSE

 NTSE

 Banking &
Insurance

 Central Govt.
Service

 State Govt.
Services

 LAW
Entrance

 MBA
Entrance

 Railways & Metro
Services

...many more

abhyasonline.in

Course
&
Test Series

Introduction to 'C' Language - Module 6

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Usage in Loops:

- Example:

```
c Copy code  
  
for (int i = 1; i <= 5; i++) {  
    if (i == 3) {  
        continue; // Skip the rest of the loop when i equals 3  
    }  
    printf("%d\n", i);  
}
```

Output:

```
Copy code  
  
1  
2  
4  
5
```



Course
&
Test Series

Introduction to 'C' Language - Module 6

Solved Example of Break Statement

CBSE

ICSE

NTSE

Banking & Insurance

Central Govt. Service

State Govt. Services

LAW Entrance

MBA Entrance

Railways & Metro Services

...many more

abhyasonline.in

`break` Statement

Example 1: Exiting a Loop

```
c Copy code
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++) {
        if (i == 6) {
            break; // Exit the loop when i equals 6
        }
        printf("i = %d\n", i);
    }
    return 0;
}
```

Output:

```
css Copy code
i = 1
i = 2
i = 3
i = 4
i = 5
```

Explanation:

- The loop runs from 1 to 10, but the ``break`` statement causes it to exit when ``i`` equals 6, so only the numbers 1 to 5 are printed.

Explanation:

- The ``break`` statement exits the inner loop when ``i * j`` equals 4. The outer loop continues with the next iteration.



Course
&
Test Series

Introduction to 'C' Language - Module 6

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Breaking out of a Nested Loop

```
c Copy code
#include <stdio.h>

int main() {
    for (int i = 1; i <= 3; i++) {
        for (int j = 1; j <= 3; j++) {
            if (i * j == 4) {
                break; // Exit the inner loop when the product equals 4
            }
            printf("i = %d, j = %d\n", i, j);
        }
    }
    return 0;
}
```

Output:

```
css Copy code
i = 1, j = 1
i = 1, j = 2
i = 1, j = 3
i = 2, j = 1
i = 2, j = 2
i = 3, j = 1
```

Explanation:

- The `break` statement exits the inner loop when `i * j` equals 4. The outer loop continues with the next iteration.

Course
&
Test Series

Introduction to 'C' Language - Module 6

Solved Example of GOTO Command

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Simple Usage of `goto`

```
c Copy code
#include <stdio.h>

int main() {
    int i = 1;

    start:
    printf("%d\n", i);
    i++;
    if (i <= 3) {
        goto start; // Jump back to the label 'start'
    }

    printf("Loop finished\n");
    return 0;
}
```

Output:

```
vbnet Copy code
1
2
3
Loop finished
```

Explanation:

- The `goto` statement sends control back to the `start` label, effectively creating a loop that runs until `i` is greater than 3.

Course
&
Test Series

Introduction to 'C' Language - Module 6

Solved Example of Continue Command

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Example 1: Skipping an Iteration

```
c Copy code
#include <stdio.h>

int main() {
    for (int i = 1; i <= 5; i++) {
        if (i == 3) {
            continue; // Skip the rest of the loop when i equals 3
        }
        printf("%d\n", i);
    }
    return 0;
}
```

Output:

```
Copy code
1
2
4
5
```

Explanation:

- The `continue` statement skips the rest of the loop body when `i` equals 3, so 3 is not printed.

Course
&
Test Series

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Introduction to 'C' Language - Module 6

Example 2: Continue in a While Loop

```
c Copy code  
  
#include <stdio.h>  
  
int main() {  
    int i = 0;  
  
    while (i < 5) {  
        i++;  
        if (i == 3) {  
            continue; // Skip the rest of the loop when i equals 3  
        }  
        printf("%d\n", i);  
    }  
    return 0;  
}
```

Output:

```
Copy code  
  
1  
2  
4  
5
```

Explanation:

- The `continue` statement skips the iteration when `i` equals 3, so 3 is not printed.