

The Break Statement

The break statement terminates the execution of the nearest enclosing do, for, switch, or while statement in which it appears. Control passes to the statement that follows the terminated statement.

Syntax

```
jump-statement :  
    break;
```

The break statement is frequently used to terminate the processing of a particular case within a switch statement. Lack of an enclosing iterative or switch statement generates an error.

Within nested statements, the break statement terminates only the do, for, switch, or while statement that immediately encloses it. You can use a return or goto statement to transfer control elsewhere out of the nested structure.

This example illustrates the break statement:

```
for ( i = 0; i < LENGTH; i++ ) /* Execution returns here when */  
{                               /* break statement is executed */  
    for ( j = 0; j < WIDTH; j++ )  
    {  
        if ( lines[i][j] == '\0' )  
        {  
            lengths[i] = j;  
            break;  
        }  
    }  
}
```

The example processes an array of variable-length strings stored in lines. The break statement causes an exit from the interior for loop after the terminating null character ('\0') of each string is found and its position is stored in lengths[i].

The variable j is not incremented when break causes the exit from the interior loop. Control then returns to the outer for loop. The variable i is incremented and the process is repeated until i is greater than or equal to LENGTH.