

### Goto Statement

C supports a unique form of a statement that is the *goto* Statement which is used to branch unconditionally within a program from one point to another. Although it is not a good habit to use the *goto* statement in C, there may be some situations where the use of the *goto* statement might be desirable.

The *goto* statement is used by programmers to change the sequence of execution of a C program by shifting the control to a different part of the same program.

'*goto*' Statement in C language

*goto* is a jumping statement in c language, which transfer the program's control from one statement to another statement (where label is defined).

*goto* can transfer the program's within the same block and there must a label, where you want to transfer program's control.

### **Defining a label**

Label is defined following by the given syntax

#### **label\_name:**

- label\_name should be a valid identifier name.
- : (colon) should be used after the label\_name.

Transferring the control using '*goto*'

Program's control can be transfer following by the given syntax

**goto label\_name;**

### **Two styles of '*goto*' statement**

We can use *goto* statement to transfer program's control from **down to top** (↑) and **top to down** (↓).

### **Transferring the control from down to top**

label-name:

statement1;

statement2;

..

**if**(any-test-condition)

goto label-name;

Here, if any-test-condition is true, *goto* will transfer the program's control to the specified label-name.

### **Example1 : To print numbers from 1 to 10 using goto statement**

```
#include <stdio.h>
void main()
{
    int number;
    number=1;
top:
    printf("%d\n",number);
    number++;
    if(number<=10)
        goto top;
}
```

## Output:

```
1
2
3
4
5
6
7
8
9
10
```

### Transferring the control from top to down

statements;

```
if(any-test-condition)
    goto label-name;
statement1;
statement2;
label-name:
    Other statements;
```

Here, if any-test-condition is true, goto will transfer the program's control to the specified label-name.

### Example 2: To read and print the number, if number is positive only.

```
#include <stdio.h>
void main()
{   int number;
    printf("Enter an integer number: ");
    scanf("%d",&number);
    if(number<=0)
        goto end;
    printf("Number is : %d", number);
end:
    printf("Bye Bye !!!");
}
```

### Output

```
First run:
Enter an integer number: 123
Number is : 123
Bye Bye !!!
```

```
Second run:
Enter an integer number: 0
Bye Bye !!!
```

### Example 3: Program to check which digit number is given by the user using goto statement.

```
#include <stdio.h>
#include <conio.h>
main()
{
    int x,i=0;
    clrscr();
    printf("enter any digit no.");
    scanf("%d",&x);
```

```

abc:
if(x!=0)
{
i++;
x/=10;
goto abc;
}
else
printf("number=%d Digit No.",i);
getch();
}

```

**Example 4: Program to print the factorial value of any number using goto keyword**

```

#include<stdio.h>
#include<conio.h>
void main()
{
int x,y=1;
clrscr();
printf("Enter any number \n");
scanf("%d",&x);
fact:
y=y*x;
if(x>1)
{
x--;
goto fact;
}
printf("%d",y);
getch();
}

```

**Try Yourself:**

1. Write a program to input any number and print the table of that number using goto statement.
2. Write a program to input any digit number and print the reverse of that digit using goto statement.
3. Write a program to print the sum of one to nth numbers using goto statement.