

CONTROL OF FLOW OF C PROGRAMMING - II

Break statement

Format of this statement is

```
break;
```

Sometimes it becomes necessary to terminate a loop depending on a condition. In such circumstance, a break statement can be used. This causes the program to exit the loop and execution continues with the statements following the loop. In case of a nested loop, only the loop in which the break statement occurs is terminated. *Break* statements interrupt the 'structured' control-flow of the program and should be avoided if possible.

Continue statement

Format of this statement is

```
continue;
```

Unlike a *break* statement, this causes the loop in which it occurs to continue. All the statements following the continue statement are ignored. This is used to bypass a group of statements in a loop based upon certain criteria or conditions. Same as the *break* statement *continue* should be avoided if possible.

if Statement

Decision making capabilities is possible with an *if* statement. The format of this statement is

```
if (condition)
{
    program statements;
}
```

The *if* statement will first examine the condition which is nothing but another boolean expression. If the expression evaluates to TRUE then the program statement following the *if* statement is executed. If the expression evaluates to FALSE then the statement following the *if* statement is skipped. A compound statement following an *if* statement has to be enclosed within a pair of curly braces.

Program 5.4

```
/* Program to calculate the absolute value of a number */
#include <stdio.h>

main()
{
    int number;
    printf("Enter a number \n");
    scanf("%d", &number);
    if(number < 0)
        number = -number;
    printf("Absolute value of the number is %d", number);
}
```

The expression evaluated by an *if* statement can be a simple boolean comparison of two variables or it can be the concatenation of several

simple boolean expressions joined by either logical AND or logical OR operators.

Program 5.5

```
/* Program to find a leap year */
#include <stdio.h>
main()
{
    int rem4; /* remainder obtained after dividing year by
4*/

    int rem100; /* remainder obtained after dividing year
by 100 */

    int rem400; /* remainder obtained after dividing year
by 400*/

    printf("Enter a year to verify\n");
    scanf("%d", &year);

    rem4 = year % 4;

    rem100 = year % 100;

    rem400 = year % 400;

    if (rem4 == 0 && rem100 == 0 && rem400 == 0)
        printf("It is a leap year !\n");
}
```

if else Statement

Another variation of an *if* statement is an *if else* statement. This statement helps a programmer to evaluate a condition and take two different actions depending on whether it results a result of TRUE or FALSE.

Program 5.6

```
#include <stdio.h>
main()
{
    int number;
    printf("Enter a number other than 0 to compute a cube:
\n");
    scanf("%d", &number);

    if (number)
        printf("Cube of %d is %d", number, pow(3,
number));
    else
        printf("Enter a valid positive integer. \n");
}
```

Source : <http://www.peoi.org/Courses/Coursesen/cprog/frame5.html>