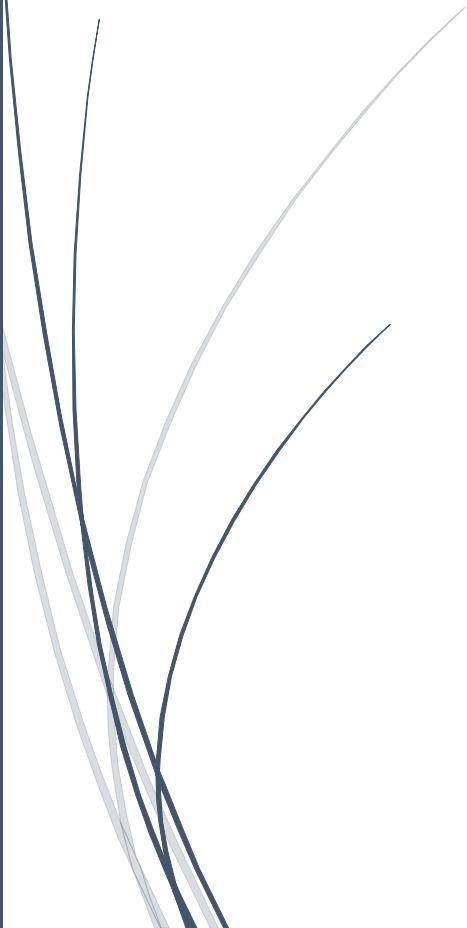


Course No. : EE207



# ***EE207-Computer Programming***

Module 2 note



## Module 2: Control statements in C

### Syllabus :-

#### Control statements in C

- ❖ if, if-else
- ❖ while, do-while
- ❖ for statements, switch
- ❖ break, continue
- ❖ go to, and labels.
- ❖ Programming examples.

- ✓ 7 Hours for this module(for 1<sup>st</sup> teaching)
- ✓ In semester exam 15% marks from this module
- ✓ In this module we studying control statements in C
- ✓ Reference ANSI C- Second edition

### Decision making and branching

C language possesses mainly 4 decision-making statements

1. If statement (simple if)
2. Switch statements
3. Conditional operators
4. go to statement

These statements are known as decision-making statements. Since these statements control the flow of execution, they are also known as control statements

### Decision making with if statements

“if” statement is a two-way decision statements and is used in conjunction with an expression. It takes the following form

#### if (test expression)

It allows the computer to evaluate the expression first and then, depending on whether the value of the expression is “true” or “false”. It transfer the control to particular statement.

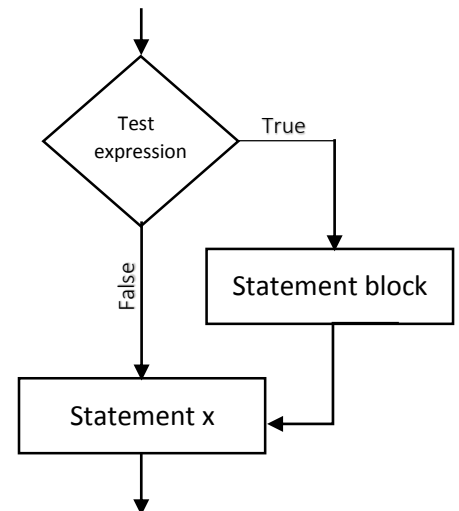
The if statement may be implemented in different forms depending on the complexity of condition to be tested. The different forms are.

- Simple ‘if’ statement
- ‘if-else’ statement
- Nested ‘if-else’ statement
- ‘else-if’ ladder

❖ **Simple if statement**

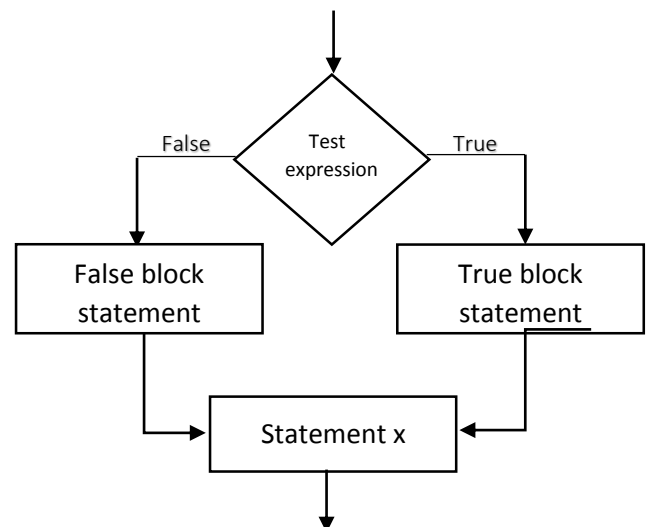
If we need to check only one condition, then we use simple If statement, it's general form & flowchart is

```
if(test condition)
{
    Statement- block
}
Statement x
```

❖ **'if else' statement**

If we want to check two conditions in a program, if-else statements are used. If else statement is an extension of the 'simple if' statement. it's general form & flowchart is

```
if (test expression)
{
    true-block statement;
}
else
{
    false-block statement;
}
statement x
```

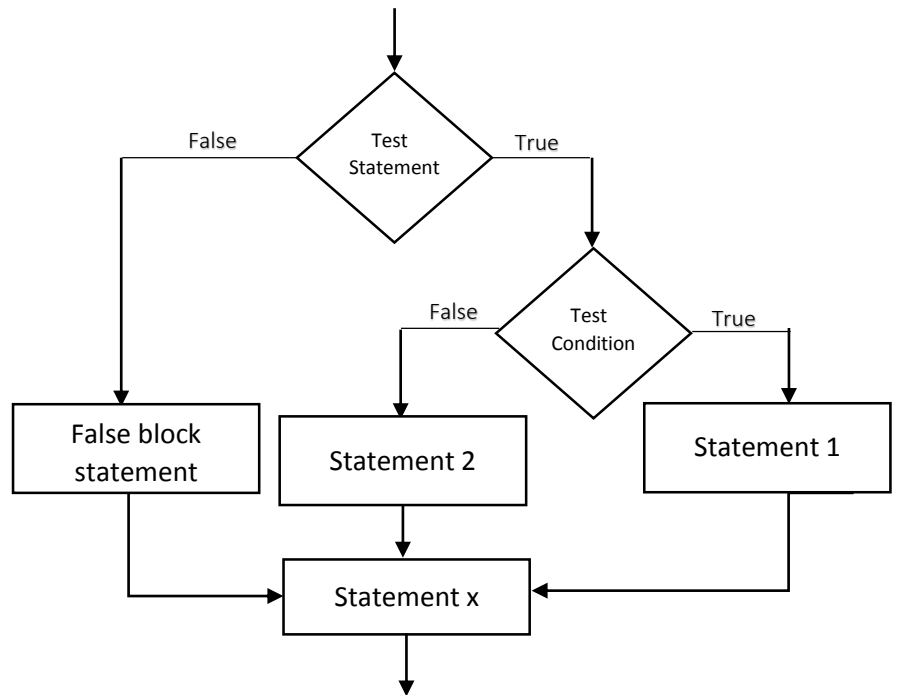
❖ **Nesting of if-else statement**

When a series of decisions are involved, we may have to use more than one 'if-else' statements in nested form. it's general form & flowchart is

```

if (test statement)
{
    if (test condition)
    {
        Statement1;
    }
    else
    {
        statement 2;
    }
}
else
{
    statement 3;
}
Statement x;

```



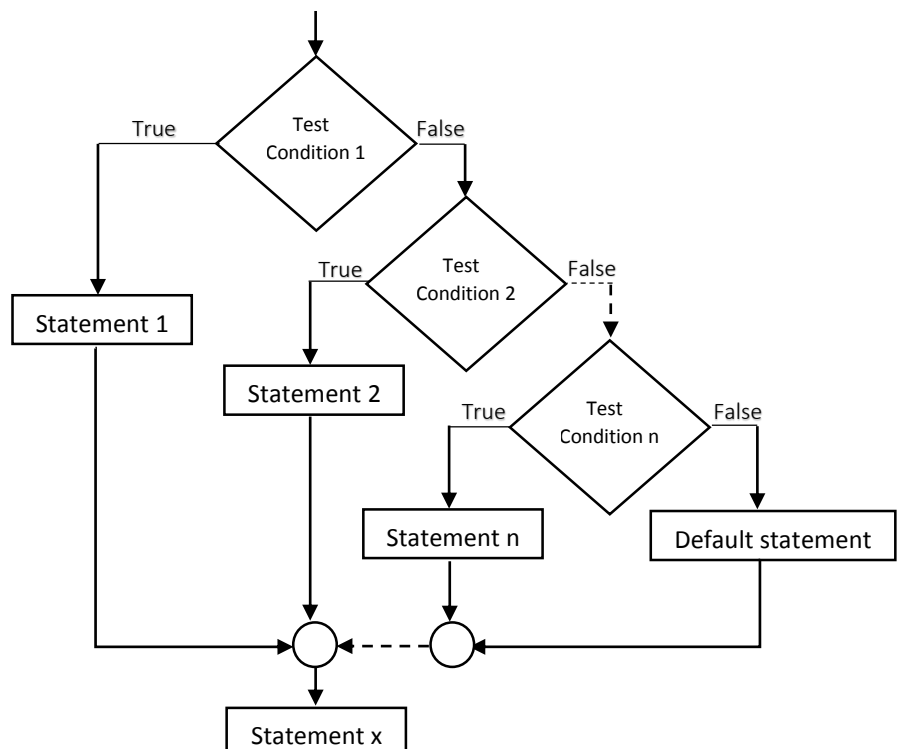
### ❖ else-if ladder

Using 'else-if' ladder is another way of putting if's together when multipath decisions are involved. If want to check more than two conditions in a program, 'else-l' statements are used. it's general form & flowchart is

```

if (test condition 1)
    Statement 1;
else if (test condition 2)
    Statement 2;
"      "
"      "
else if (test condition n)
    Statement n;
else
    default statement;
statement x;

```

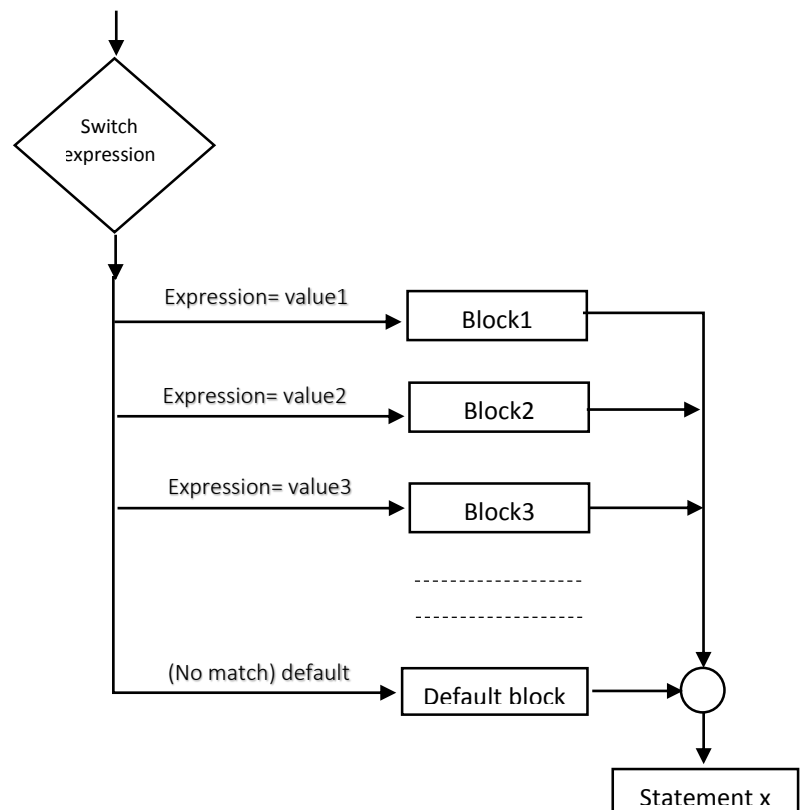


## The switch statement

If we want to select one option from more than one options (multiple option), we can use an 'if' statement to control the selection. However, the complexity of such a program increases when the number of options increases. Then the program becomes difficult to read and follow

In such cases, in order to reduce the complexity of the program, a multi-way decision statement known as 'switch' statement is used. The switch statement tests the value of the given variable against a list of case values and when a match is found, a block of statement associated with that is executed

```
switch (expression)
{ case value1 : block1;
  Break;
  case value2 : block2;
  Break;
  " "
  " "
  case value-n : blockn;
  Break;
  default : default block;
  Break;
}
Statement x;
```



## The 'goto' statement

The goto statement is used to branch unconditionally from one point to another in the program. The goto requires **label** in order to identify the place where the branch is to be made. A label any valid variable name, and must followed by a colon.

### Forward jump

```
goto label; _____
.....
.....
.....
label: statement; ←
```

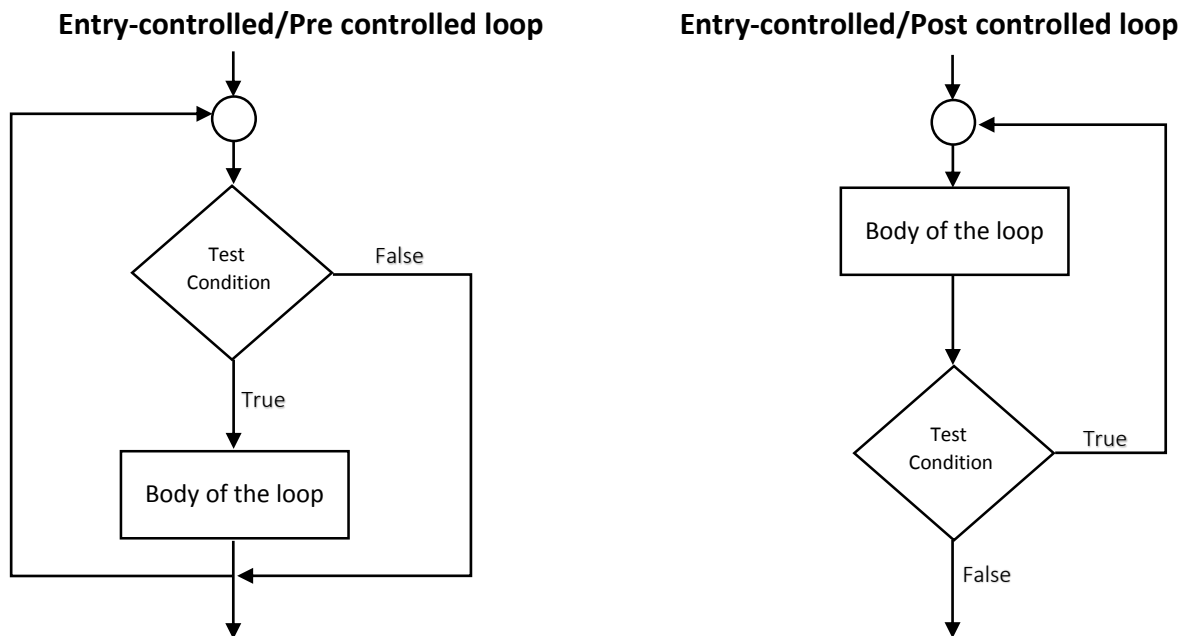
### Backward jump

```
label: statement; ←
.....
.....
.....
goto label; _____
.....
.....
```

## Looping

In looping, a sequence of statements are executed until conditions for termination of the loop are satisfied. A program loop therefore consists of two segments, one known as **body of the loop** and other known as the **control statement**.

Depending on the position of the control statement in the loop, a control structure may be classified either as the **entry-controlled loop** or as the **exit-controlled loop**.



The C language provides three constructs for performing loop operations. They are

1. The 'while' statement
2. The 'do' statement
3. The 'for' statement

### While statement

- Simplest of all looping structure
- General form

```
while (test condition)
{
    body of the loop
}
```

**'do' statement**

- For loop is an exit-controlled loop
- General form

```
do
{
    body of the loop
}
while (test condition);
```

**'for' statement**

- For loop is an entry-controlled loop
- General form

```
For (initialization;testcondition;increment)
{
    body of loop;
}
```

**Jumps in loops**

Jumping out of a loop can be done using 2 statements:-

- ✓ break
- ✓ goto

when a 'break' statement is encountered inside a loop, the loop is immediately exited and the program continues with the statement immediately following the loop

since a goto statement can transfer the control to any place in a program, it is very useful to provide branching within a loop.

Eg;-

```

while(.....)
{
    .....
    .....
    .....

    if(condition)
        break;
    .....
    .....
}

```

while(.....)

```

{
    .....
    .....
    If(error)
        Goto stop;
    .....
    .....
    if(condition)
        goto abc;
    .....
    .....
    abc :
    .....
}

Stop: .....
.....

```

### Skipping a part of loop

- **Continue** statement

Unlike the break which causes the loop to be terminated, the 'continue' statement as the name implies causes the loop to be continued with next iteration after skipping any statement in between

Eg;-

```

While(test condition)
{
    .....
    If (condition)
        Continue;
    .....
    .....
}

```

Note prepared by

VISHNU CV KTU live  
vishnucvd@gmail.com

For more notes and question papers visit KTU live  
All module Notes of Computer programming now available at our site



Now we available at WhatsApp 81-57097-880 (Send ur name reg. no & name to this to connect with us)