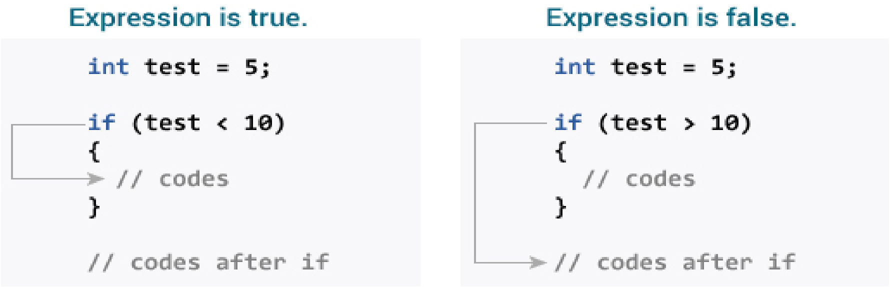
AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

CHAPTER - 3
CONTROL STUCTURE IN C


Expression is true.

```
int test = 5;

if (test < 10)
{
   // codes
}

// codes after if
```

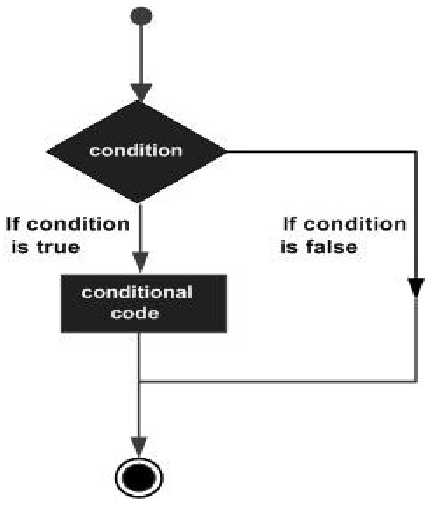
Expression is false.

```
int test = 5;

if (test > 10)
{
   // codes
}

// codes after if
```

Subject : PPS **Code : 3110003**	**Prepared By:** **Asst. Prof. Rupali Patel** **(CSE Department, ACET)**	AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

Block Diagram of Computer

Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.



AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

if statement in C Programs

The if statement facilitates to check a particular condition. If that condition is true, then a specific block (enclosed under the if) of code gets executed.

Syntax:

```
if(condition)
{
        //Statement block
}
```

if-else Statement

The if statement works pretty good, but if you want to work with more variables and more data, the if-else statement comes into play.
In the if statement, only one block of code executes after the condition is true.
But in the if-else statement, there are two blocks of code – one for handling the success and other for the failure condition.

Syntax:

```
if(condition)
{
   //Statement block
}
else
{
   //Statement block
}
```

nested if-else Statement

It is always legal in C programming to **nest** if-else statements, which means you can use one if or else if statement inside another if or else if statement(s).

nested if-else Statement (cont..)

Syntax:

```
if(condition)
{
        if(condition)
        {
                //Statement block
        }
        else
        {
                //Statement block
        }
}
else
{
        //Statement block
}
```

Switch Statement

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case**.

Syntax:

```
switch(expression)
{
  case constant-expression  :
                                                  statement(s);
                                                  break; /* optional */
   default : /* Optional */
                                                  statement(s);
}
```

The ? : Operator

We have covered **conditional operator ? :** in the previous chapter which can be used to replace **if...else** statements. It has the following general form −
Exp1 ? Exp2 : Exp3;

Where Exp1, Exp2, and Exp3 are expressions. Notice the use and placement of the colon.

The value of a ? expression is determined like this −
Exp1 is evaluated. If it is true, then Exp2 is evaluated and becomes the value of the entire ? expression.

If Exp1 is false, then Exp3 is evaluated and its value becomes the value of the expression.

Looping Statement in C

Looping statement are the statements execute one or more statement repeatedly several number of times. In C programming language there are three types of loops; while, for and do-while.

Advantage with looping statement
Reduce length of Code
Take less memory space.
Burden on the developer is reducing.
Time consuming process to execute the program is reduced.

Types of Loops.
There are three type of Loops available in 'C' programming language.
while loop
for loop
do..while

While loop

In **While Loop in C** First check the condition if condition is true then control goes inside the loop body other wise goes outside the body. **while loop** will be repeats in clock wise direction.

Syntax:

```
Assignment;
while(condition)
{
Statements;
......
Increment/decrements (++ or --);
}
```

For loop

For Loop in C is a statement which allows code to be repeatedly executed. For loop contains 3 parts Initialization, Condition and Increment or Decrements.

Syntax:

```
for(initialization; condition; increment/decrement)
{
Statements;
}
```

do-while loop

A do-while Loop in C is similar to a while loop, except that a do-while loop is execute at least one time.
A do while loop is a control flow statement that executes a block of code at least once, and then repeatedly executes the block, or not, depending on a given condition at the end of the block (in while).

Syntax:

```
do
{
Statements;
........
Increment/decrement (++ or --)
} while(condition);
```

break statement

In C programming, break statement is used with conditional if statement. The break is used in terminating the loop immediately after it is encountered. it is also used in switch...case statement.

Syntax:
break;

continue statement

It is sometimes desirable to skip some statements inside the loop. In such cases, continue statement is used.

Syntax:

continue;

Just like break, continue is also used with conditional if statement.

goto statement

In C programming, goto statement is used for altering the normal sequence of program execution by transferring control to some other part of the program.

Syntax:

```
goto label;
.............
.............
.............
label:
statement;
```

In this syntax, **label** is an identifier.

Thank you