

Jay Shree Krishna



**RUPAREL
EDUCATION PVT. LTD.**

CIN : U80300GJ2013PTC075575

WELCOME TO

RUPAREL EDUCATION PVT. LTD.

201,Punit Shopping Centre
Ranavav Chowk
Junagadh.
Mobile No.76000 44051.

Prepared By : Uday Shah

201, Punit Shopping Center, M.G.Road, Junagadh. M. 76000 440 51, 98255 36800.

:: DECISION CONTROL STRUCTURE ::



**RUPAREL
EDUCATION PVT. LTD.**

CIN : U80300GJ2013PTC075575

- **The if-else statement:**
- The if-else statement is used to carry out a logical test and then take one of two possible actions depending on the outcome of the test (i.e. whether the outcome is true or false).
- The else position of the if-else statement is optional. Thus, in its simplest general form, the statement can be written.
- **if(expression)**
{
statement;
}
- The expression must be placed in parenthesis, as shown. In this form, the statement will be executed only if the expression has non-zero value (i.e. true).

Prepared By : Uday Shah



- If the expression has a value of zero (i.e. expression is false), then the statement will be ignored.
- The statement can be either simple or compound. In practice, it is often compound statement, which may include other control statements.

• The general form of an if statement which include the else clause is
if(expression)

```
{  
    statement 1;  
}  
else  
{  
    statement 2;  
}
```

If the expression has a non-zero value, then statement 1 will be executed.



- Otherwise, (i.e. expression is false), statement 2 will be executed.
- It is possible to nest if...else statement within one-another, just as we did with loops, there are several different forms that nested if...else two-layer nesting is :
- In this situation, one complete if... else statement will be executed if expression is nonzero(true) and another complete if else statement will be executed if expression is false (zero).
- It is of course, possible that statement 1, statement2, statement3 and statement4 will contain another if...else statements. We would then have multi layer nesting.
- Some other forms of two layer are :

Prepared By : Uday Shah



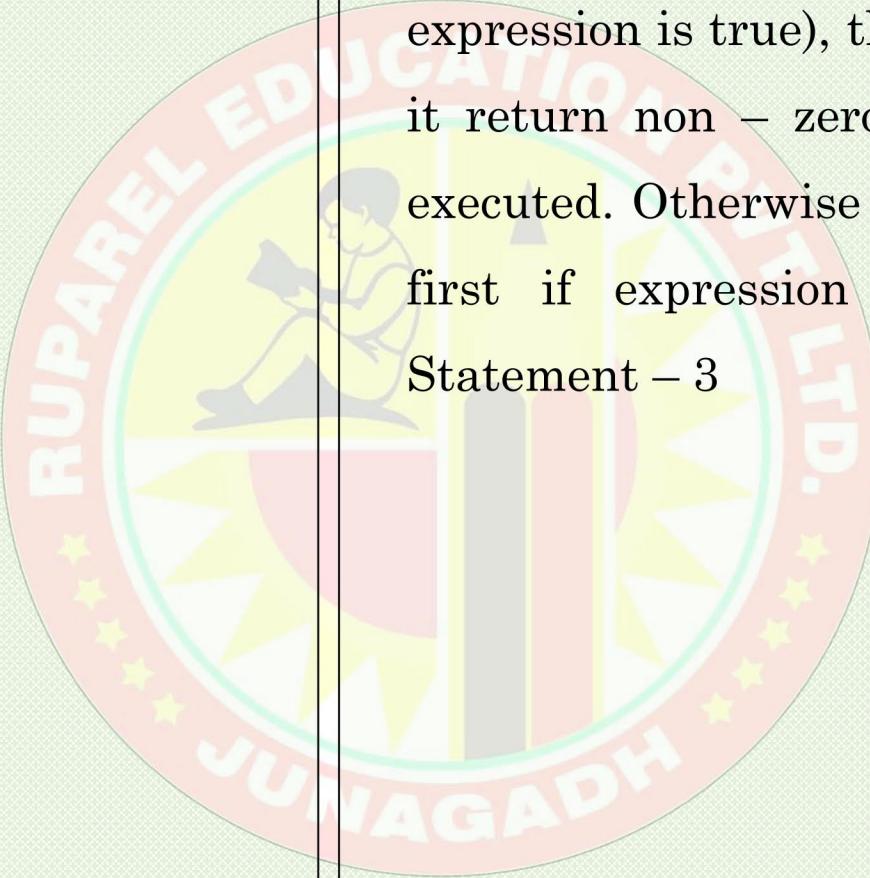
```
• if<exp1>
{
    statement1;
}
else
{
    if<exp2>
    {
        statement2;
    }
}
```

- If the first if expression has a non-zero value (i.e. if expression is true), then statement 1 will be executed. Otherwise, (i.e. expression is false), second if expression is evaluated and test in else block if it return non zero value then statement 2 will be executed.

Prepared By : Uday Shah



```
• if<exp1>
{
    if<exp2>
    {
        statement1;
    }
    else
    {
        statement2;
    }
}
else
{
    statement3;
}
```

- 
- If the first if expression has a non-zero value (i.e. if expression is true), then it check second if expression if it return non – zero value then statement 1 will be executed. Otherwise statement 2 will be executed. But first if expression has return zero then execute Statement – 3

Prepared By : Uday Shah



```
• if<exp1>
{
    if<exp2>
    {
        statement1;
    }
    else
    {
        statement2;
    }
}
else
{
    if<exp3>
    {
        Statement3;
    }
    else
    {
        Statement4;
    }
}
```

- If the first if expression has a non-zero value (i.e. if expression is true), then it check second if expression and execute statement 1 or 2 otherwise it check third if expression and execute statement 3 or 4



```
if<exp1>
{
    statement1;
}
else if <exp2>
{
    statement2;
}
else if<exp3>
{
    Statement 3;
}
else
{
    Statement 4;
}
```

- If the first if expression has a non-zero value then it execute statement 1 otherwise it check second if expression and execute statement 2 otherwise execute last if expression and execute statement 3 if all expression are false then it execute statement 4.

Prepared By : Uday Shah

Switch Case Statement



- The switch case statement provide and alternative way to tack decision it is work on user choice.
- It means it provide a way by which user can tack their on decision.
- It is not a logic but provide powerful structure to maintain more than one logic in it.
- Switch case statement tack an integer expiration for execute a specific block.
- In switch case statement each case is ~~printed~~ ^{Treated} label.
- Switch statement have a default option if none of the expiration match with any case than it execute default case.
- Following is a syntax of switch case statement.

Prepared By : Uday Shah



Switch(expression)

{

Case 1:

 Statement;

 Break;

Case 2:

 Statement;

 Break;

Default:

 Statement;

}

According to the syntax switch is reserve keyboard and it contain expression in character or in int type data
All the case are contain within 2 curly bracket and each case has separate label.
Switch case statement contain break in every case statement to stop execution of switch case after maintain their desire case.
Default case statement does not require break statement because it always written at last in switch case block.

Prepared By : Uday Shah

Conditional Ternary Statement



**RUPAREL
EDUCATION PVT. LTD.**

CIN : U80300GJ2013PTC075575

- Conditional or the ternary statement can be use in replace to the statement of if else in a some situation the conditional operator(?) is unique in a C-Language.
- The ternary operator manse the operator contain three operands.
- The first operands is always evaluated first.
- It is usually conditional excretion that use the logical operators.
- The next two operands are evaluated biased on their condition if condition is true than it execute second operand otherwise execute third operand.
- **Following is syntax of conditional ternary operator.**
 - Conditional expiration ? expiration 1: expiration 2;
- According to above syntax conditional expiration contain either relational or logical part and explain one and two are evaluated beside on there conditional expiration.

Prepared By : Uday Shah

Looping Structure



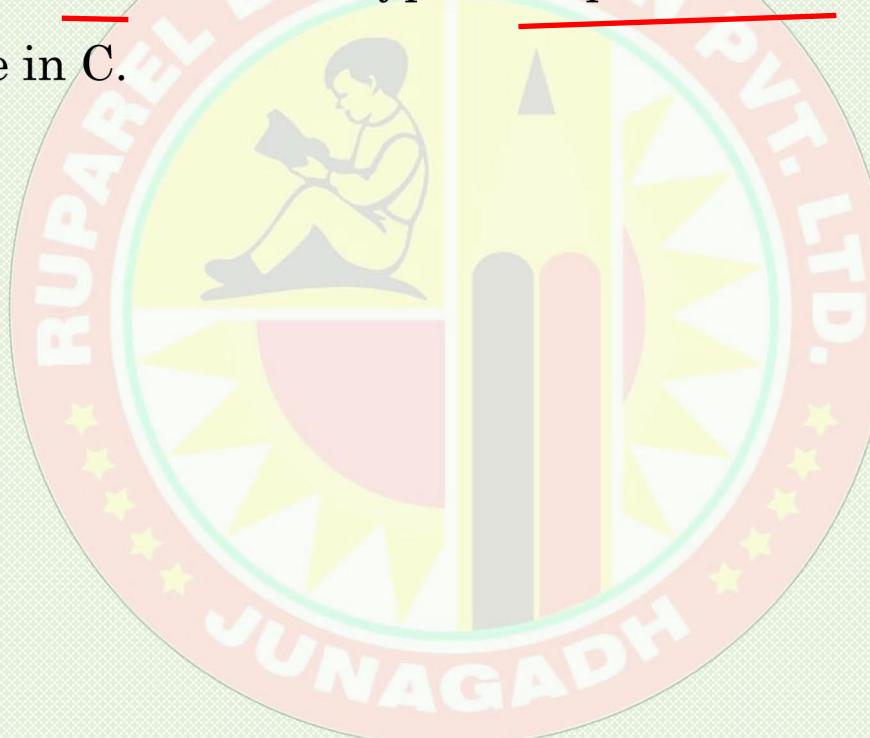
RUPAREL
EDUCATION PVT. LTD.

CIN : U80300GJ2013PTC075575

• LOOP CONTROL STRUCTURE

If we want to perform certain action for no of times or we want to execute same statement or a group of statement repeatedly then we can use different type of loop structure available in C. Basically there are 3 types of loop structure available in C.

- (1) While loop ✓
- (2) Do... while ✓
- (3) For loop ✓



Prepared By : Uday Shah



- **While statement :**

The while statement is used to carry out looping operations. The general form of the statements

- **initialization;**

```
while(exp)
```

```
{
```

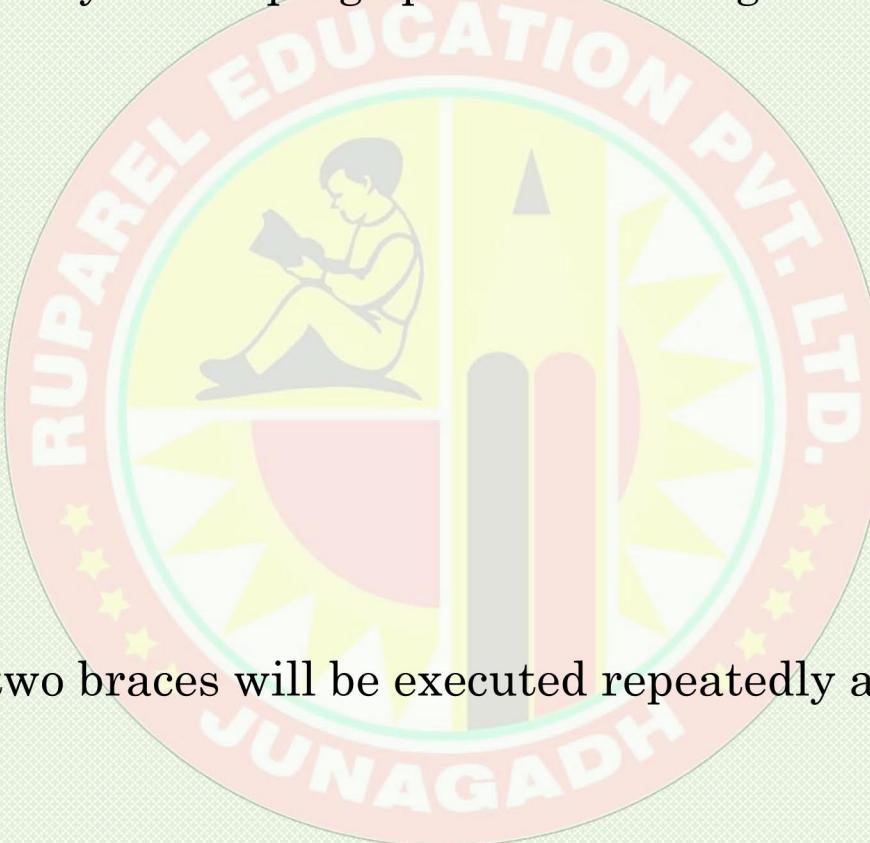
```
statement 1;
```

```
statement 2;
```

```
increment/ decrement;
```

```
}
```

- The enclose statements within two braces will be executed repeatedly as long as the expression evaluated as true.



Prepared By : Uday Shah



- When the expression is evaluate a false or when condition will be false then it will come out from the loop and stop the execution of that loop.
- Initialization statement initialize some memory variable with some value.
- Increment or decrement operator increase or decrease the value of operator is use by expression. This statement can be simple of compound, though it is typically a compound statement, it must include some features, which eventually offers the value of expression, thus provides a stopping condition for the loop.
- In the practice, the included expression is usually a logical expression that is either true or false. Remember that true corresponds to non-zero value and false corresponds to zero value.

Prepared By : Uday Shah



- Thus the statement will continue to execute as long as the logical expression is true.
- The part of while statement, which contains the statement, is called the body of the loop.
- Body of the loop may have one or more statements.
- The braces are needed only if the body contains two or more statements.
- However, it is a good practice to use braces even if the body has only one statement.
- Suppose we want to display the consecutive digits 0,1,2,3,...9 with one digit on the each line.
- This can be accomplished with the following program

Prepared By : Uday Shah



```
• #include<stdio.h>

main ()
{
    int digit = 0;
    while(digit<=9)
    {
        printf("%d \n",digit);
        digit++;
    }
}
```

According to example

Initially, digit is assigned a value of 0.

The while loop then displays the current value of digit, increases its value by 1

and then repeats the cycle, until the value of the loop will be repeatedly 10

times, resulting in 10 consecutive lines of output.

Each line will contain a successive integer value, beginning with and ending with 9.

Thus when the program to run, the following output will be generated :

0 1 2 3 4 5 6 7 8 9

Prepared By : Uday Shah

Do-While Statement



**RUPAREL
EDUCATION PVT. LTD.**

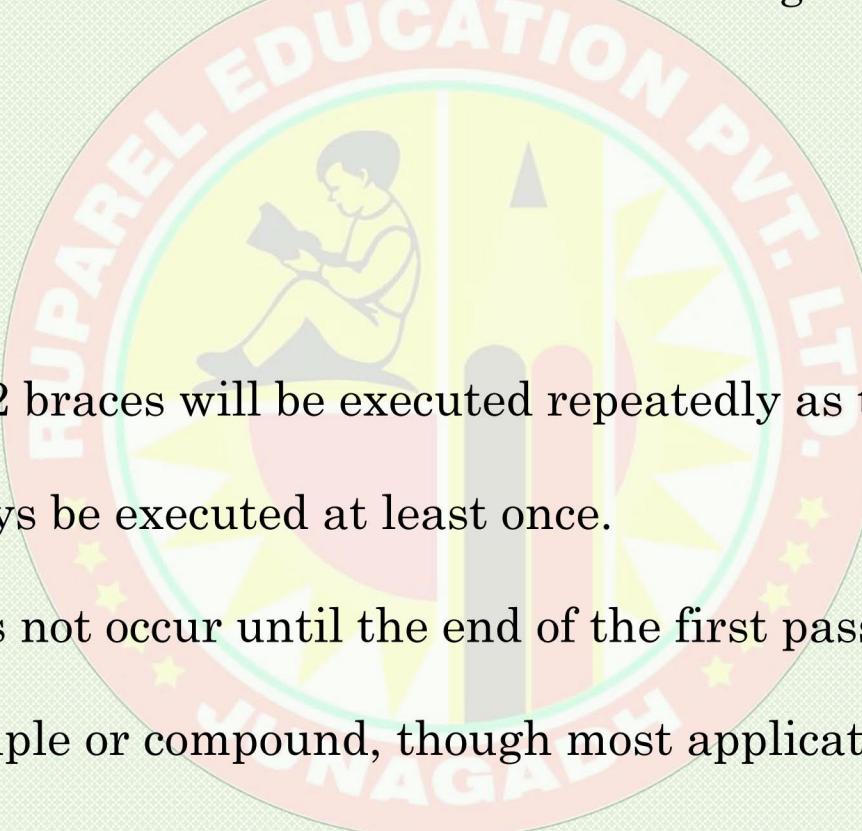
CIN : U80300GJ2013PTC075575

- Sometimes, however, it is desirable to have a loop with the test for continuation at the end of each pass.

This can be accomplished by means of the do-while statement. The general form of do-while statement is

```
• do  
{  
    statement 1;  
    statement 2;  
    increment/ decrement;  
} while (exp);
```

- The enclosed statement within 2 braces will be executed repeatedly as the value of expression is not zero.
- Notice that statement will always be executed at least once.
- Since the test for repetition does not occur until the end of the first pass through the loop.
- The statement can be either simple or compound, though most applications will require it to be a compound statement.



Prepared By : Uday Shah



- It must include some feature which eventually alters the value of expression so that the looping action can terminate.
- Since the expression is evaluated at the bottom of the loop, the do...while loop construct provides an exit... controlled loop and therefore the body of the loop is always executed at least once.
- In practice, expression is usually a logical expression, which is either true(with a non zero value) or a false(with a value of 0).
- The included statement will be repeated if the logical expression is true.
- For most applications it is more natural to test for continuation of a loop at the beginning rather than at the end of the loop for this reason, the do...while statement is used less frequently than while statement.
- Suppose that we want to display consecutive 0,1,2,3,...9 with one digit on each line. This can be accomplished with the following program

Prepared By : Uday Shah



```
• #include<stdio.h>

main ()
{
    int digit = 0;
    do
    {
        printf("%d \n",digit);
        ++digit ;
    } while(digit<=9);
}
```

The digit is initially assigned a value of 0.
The do while loop displays the current value of digit.
Increases its value by 1, and then tests to see if the current value of digit exceeds 0. If so loop terminates; otherwise the loop continues, using the new value of digit.
Note that the last is carried out at the end of each pass through the loop.
The net effect is that the loop repeats 10 times, resulting in 10 successive lines of output.
Thus, when the program is run the following output is generated :

0 1 2 3 4 5 6 7 8 9

Prepared By : Uday Shah

FOR Statement



**RUPAREL
EDUCATION PVT. LTD.**

CIN : U80300GJ2013PTC075575

- The for statement is another entry controller that provides a more concise loop control structure. The general form of the for loop is
- **for(initialization; test condition; increment)**
{
 statement 1;
 statement 2;
}
- The execution of the for statement is as follows:
 1. Initialization of the control variables is done first, using assignments statement such as `I=0` and `count=0`. The variables `I` and `count` are known as loop controls.

Prepared By : Uday Shah



- 2. The value of the control variables is tested using the test condition. The test condition is relational expression, such as $i > 0$ or $i < 10$ that determines when the loop is terminated and the execution continues with statement that immediately follow by the loop.
- 3. When the body of the loop is executed, the control is transferred back to the for statement either evaluating the last statement in the loop.
- Now, the control variable is incremented using and assignment statement such as $i = i + 1$ and if the new value of the control is satisfied under test condition, then the body of the loop is executed.
- This process continues till the value of the control variable fails to satisfy the test-condition.

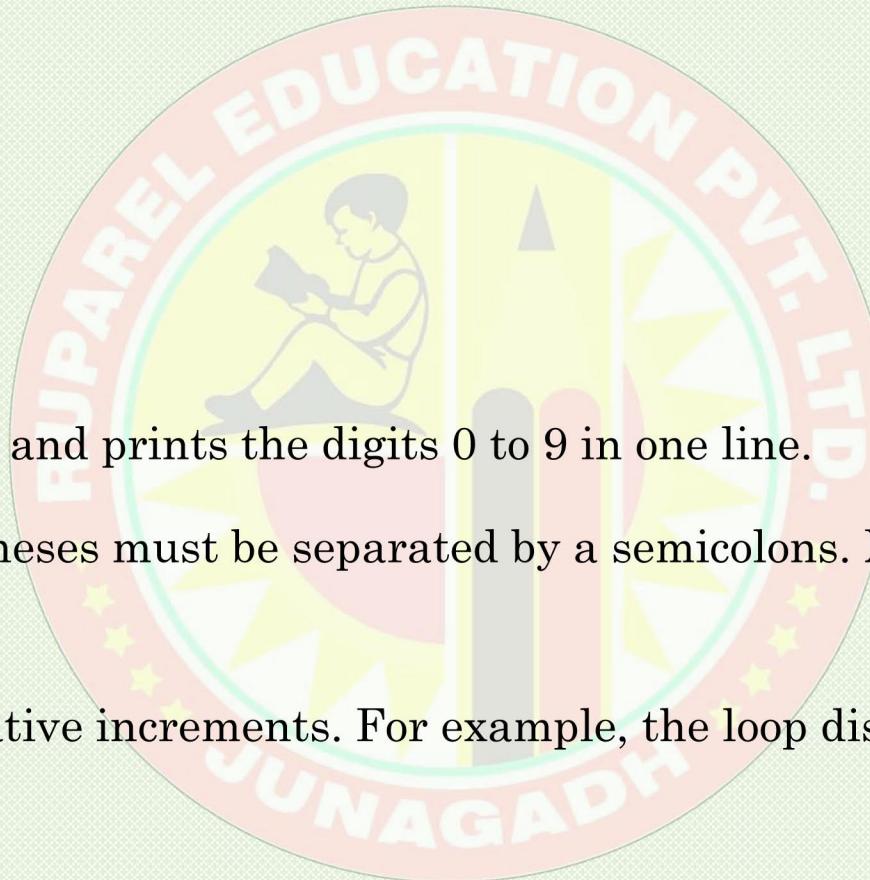
Prepared By : Uday Shah



- (a) Consider the following segment of a program

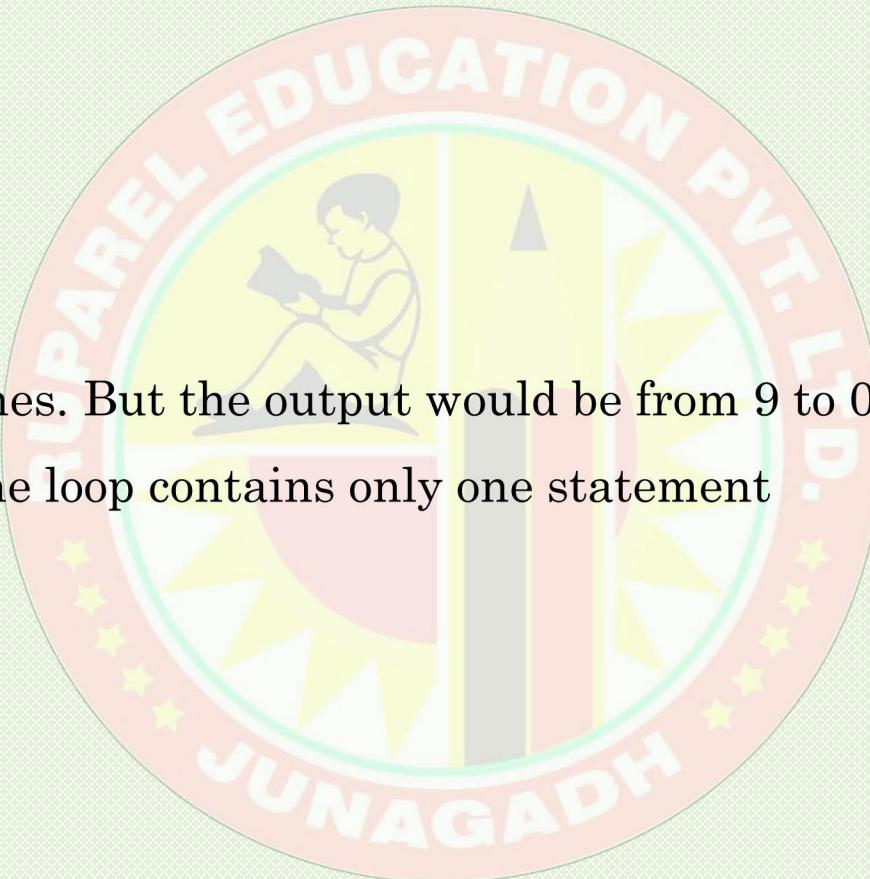
```
• for(i=0; i<9; i++)  
{  
    printf("%d", i);  
    printf("\n");  
}
```

- This for loop is executed 10 times and prints the digits 0 to 9 in one line.
- The three sections within parentheses must be separated by a semicolons. Note that there is no semicolon at the end of the increment section i++.
- The for statement allows for negative increments. For example, the loop discussed above can be written as follows





- ```
for(i=9; i>=0; i--)
{
 printf("%d",i)
 printf("\n");
}
```
- This loop is also executed 10 times. But the output would be from 9 to 0 instead of 0 to 9. Note that, braces are optional when the body of the loop contains only one statement



Prepared By : Uday Shah

# Jumping statements



**RUPAREL  
EDUCATION PVT. LTD.**

CIN : U80300GJ2013PTC075575

## • BREAK STATEMENT

- The break statement is used to terminate loops or to exit a switch.
- The break statement will break or terminate the inner-most loop. It can be used within a while, a do-while, a for or a switch statement.
- The break statement is written simply as break; Without any embedded expressions or statement. For example :

```
for(i=1; i<=10; i++)
{
 if(i==5)
 break;
 printf("\ni=%d",i);
}
```
- The output will be 1,2,3,4 and then break will terminate this loop and stop the execution of the for loop.

Prepared By : Uday Shah

# CONTINUE STATEMENT

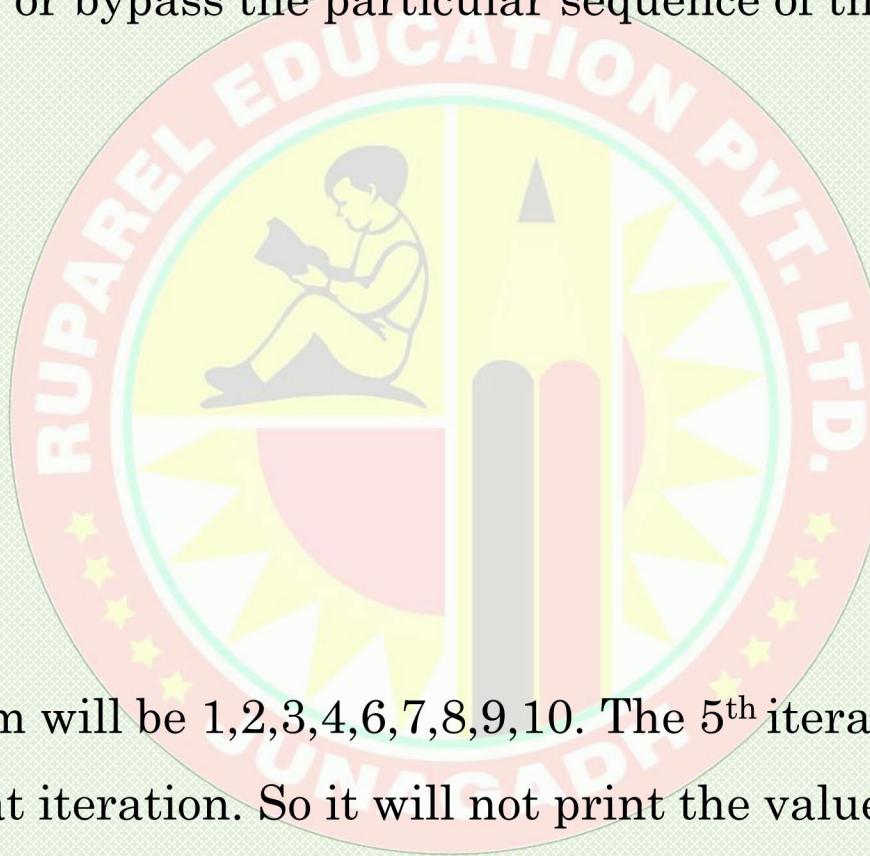


**RUPAREL  
EDUCATION PVT. LTD.**

CIN : U80300GJ2013PTC075575

- The continue statement is used to skip or to bypass some step or iteration of looping structure. It does not terminate the loop but just skip or bypass the particular sequence of the loop structure. It is simply written as continue.

```
for(i=1; i<=10; i++)
{
 if(i==5)
 continue;
 printf("\ni=%d",i);
}
```



- The output of the above program will be 1,2,3,4,6,7,8,9,10. The 5<sup>th</sup> iteration of the loop will be skipped as we have define the continue for that iteration. So it will not print the value '5'

Prepared By : Uday Shah

# THE GOTO STATEMENT



- The goto statement is used to alter the normal sequence of program execution by transferring control to some other part of the program.
- In its general form the goto statement is written as goto label;
- Where label is an identifier used to label the target statement to which control will be transferred.
- Control may be transferred to any other statement within the program.
- The target statement must be labeled and the label must be followed by a colon.
- Thus the target statement will appear as label: statement.
- Each labeled statement within the program must have unique label, i.e. no two statements can have same label.

Prepared By : Uday Shah



- **Syntax of goto Statement**

```
void main()
{
```

Label:

```

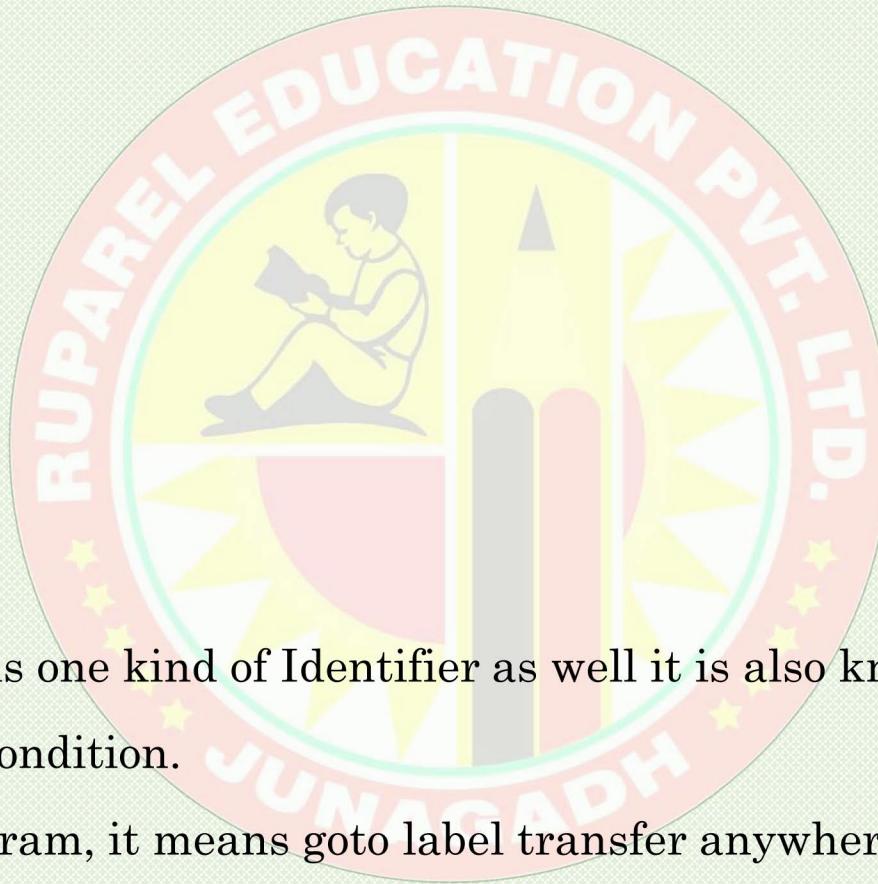


```

```
if(Condition)
{
 goto Label;
}
```

```
}
```

- According to above syntax Label is one kind of Identifier as well it is also known as flag value.
- Label is always enclosed with if condition.
- Label may be anywhere in a program, it means goto label transfer anywhere.



Jay Shree Krishna



**RUPAREL  
EDUCATION PVT. LTD.**

CIN : U80300GJ2013PTC075575

## **RUPAREL EDUCATION PVT. LTD.**

201,Punit Shopping Centre  
Ranavav Chowk  
Junagadh.  
Mobile No.76000 44051.



Prepared By : Uday Shah