# Conditional Statements

# Relational Operators: ⇒

The lists of relational operators available in **C** are:

| *Operators* | *Meanings* |
|---|---|
| = = | Equal to |
| 〉 | Greater than |
| 〈 | Less than |
| 〉= | Greater than or equal to |
| 〈= | Less than or equal to |
| != | Not equal to |

# Logical Operators: ⇒

C allows usage of the following three logical operators:

(i) && ⇒ AND
(ii) ‖ ⇒ OR
(iii) ! ⇒ NOT

# Decision control instruction:

Some times it is desired that a set of instructions to be executed in one situation and an entirely different set of instructions to be executed in another situation. This kind of situation is dealt in **C** programming using a decision control instruction. Decision control instruction can be implemented in **C** using:

(a) The ***if*** statement
(b) The ***if-else*** statement
(c) The ***condition*** operator.

(a) The ***if*** statement: →

C uses the key word '***if***' to implement the decision control instruction. The general form of ***if*** is;

```
if (this condition is true)
        execute this statement;
```

e.g., 1)
```
if (a<10)
        printf ("Thank you");
```
2)
```
if (a>=6)
        x = x+1;        etc.
```

Multiple statements with in ***if***:

It may also happen that in a program we want more than one statement to be executed if the condition following ***if*** is satisfied.

Syntax:

```
if (this condition is true )
{
        statement1;
        statement2;
        ……;
        statementn;
}
```

```
e.g.,   if (a>10)
        {
                c=c+1;
                printf ("The condition is satisfied");
        }
```

Example program-5
Filename: EP5.C

```
/* Demonstration of if statement */
main( )
{
        int a;
        printf ("Enter a number less than 10");
        scanf ("%d",&a);
        if (a < 10)
        printf ("What an obedient servant you are !");
}
```

(b) The ***if-else*** statement: →

Here a statement or group of statements will be executed if the condition is true and another statement or group of statements will be executed if the condition is not true.

Syntax:

```
if (logical expression)
{
        statement1;
        statement2;
        …….;
        statementn;
}
else
{
        statement1;
        statement2;
        …….;
        statementn
}
```

Example: 1)

```
if (x= =10)
        a = 6;
        a = 7;
```

```
2)      if (x>10)
        {
                a = 6;
                b = 7;
        }
        else
        {
                a = 7;
                b = 6;
        }
```

Nested ***if-else***:

If we write an entire ***if-else*** with in the body of the ***if*** statement or the body of the ***else*** statement then this is called 'nesting' of ***if***'s.

Syntax:

```
if (condition)
{
        if (condition)
                do this;
        else
        {
                do this;
                and this;
        }
}
```

Example program-6

Filename: EP6.C

```
/* Demonstration of nested if-else */
main( )
{
        int i;
        printf ("Enter either 1 or 2");
        scanf ("%d",&i);
        if (i= =1)
                printf ("You would go to heaven!");
        else
        {
                if (i= =2)
                        printf ("You will go to Hell!");
                else
                        printf ("No risk no gain");
        }
}
```

Different forms of ***if***:

(i)
```
if (condition)
        do this;
```

(ii)
```
if (condition)
        do this;
else
        do this;
```

(iii)
```
if (condition)
{
        do this;
        and this;
}
```

(iv)
```
if (condition)
{
        do this;
        and this;
}
else
{
        do this;
        and this;
}
```

(v)
```
if (condition)
        do this;
else
{
        if (condition)
                do this;
        else
        {
                do this;
                and this;
        }
}
```

(vi)
```
if (condition)
{
        if (condition)
                do this;
        else
        {
                do this;
                and this;
        }
}
else
do this;
```

Example program-7
Filename: EP7.C

```
/* Demonstration of logical operators */
main( )
{
        float m1, m2, m3, m4, m5, per;
        printf (“Enter marks in five subjects\n”);
        scanf (“%f%f%f%f%f”, &m1,&m2,&m3,&m4,&m5);
        per = (m1+m2+m3+m4+m5)/5;
        if (per >= 60)
                printf (“First division”);
        if ((per >=50) && (per <60))
                printf (“ Second division”);
        if (( per >=40) && (per <50))
                printf (“Third division”);
        if ( per <40)
                printf (“fail”);
}
```

Example program-7a
Filename:EP7A.C

```
/* Previous program using if-else */
main( )
{
        float m1, m2, m3, m4, m5, per;
        printf (“Enter marks in five subjects\n”);
        scanf (“%f%f%f%f%f”, &m1,&m2,&m3,&m4,&m5);
        per = (m1+m2+m3+m4+m5)/5;
        if (per >= 60)
                printf (“First division”);
        else
                if (per>=50)
                        printf (“Second division”);
                else
                        if (per >=40)
                                printf (Third division”);
                        else
                                printf (“Fail”);
}
```

(c) The ***conditional*** operator:

The conditional operators used in **C** are **?** and **:**.

The syntax is;

$expression_1$ ? $expression_2$ : $expression_3$;

If $expression_1$ is true, then the value returned will be $expression_2$ otherwise the value returned will be $expression_3$.

e.g., i) y = (x > 5 ? 3:4);

Here y will store 3 if x is greater than 5 otherwise it will store 4 in y.

ii)
```
int i;
scanf ("%d",&i);
(i = = 1 ? printf ("Amit") : printf ("Sumit");
```

Here the output will be Amit if i = 1, otherwise the output will be Sumit.

iii) a > b ? g = a : (g = b);

Here g will store a if a is greater than b otherwise g will store b.