OPERATORS AND EXPRESSIONS

ARITHMETIC OPERATORS

The basic operators for performing arithmetic are the same in many computer languages:

```
+       addition
-       subtraction
*       multiplication
/       division
%       modulus (remainder)
```

For exponentiations we use the library function **pow**. The order of precedence of these operators is % / * + - . it can be overruled by parenthesis.

Integer division :
Division of an integer quantity by another is referred to integer division. This operation results in truncation. i.e.When applied to integers, the division operator `/` discards any remainder, so `1 / 2` is 0 and `7 / 4` is 1. But when either operand is a floating-point quantity (type `float` or `double`), the division operator yields a floating-point result, with a potentially nonzero fractional part. So `1 / 2.0` is 0.5, and `7.0 / 4.0` is 1.75.

.

Example :
```
int a, b, c;
a=5;
b=2;
c=a/b;
```
Here the value of c will be 2

Actual value will be resulted only if a or b or a and b are declared floating type.
The value of an arithmetic expression can be converted to different data type by the statement (data type) expression.

Example :
```
int a, b;
float c;a=5;b=2;
c=(float) a/b
```

Here c=2.5

Order of Precedence

Multiplication, division, and modulus all have higher *precedence* than addition and subtraction. The term ``precedence" refers to how ``tightly" operators bind to their operands (that is, to the things they operate on). In mathematics, multiplication has higher precedence than addition, so `1 + 2 * 3` is 7, not 9. In other words, `1 + 2 * 3` is equivalent to `1 + (2 * 3)`. C is the same way.

UNARY OPERATORS

A operator acts up on a single operand to produce a new value is called a unary operator.
(1) the **decrement and increment** operators - ++ and -- are unary operators. They increase and decrease the value by 1. if x=3 ++x produces 4 and –x produces 2.

Note : in the place of ++x , x++ can be used, but there is a slight variation. In both csse x is incremented by 1, but in the latter case x is considered before increment.

(2) sizeof is another unary operator

```
int x, y;
y=sizeof(x);
```

The value of y is 2 . the *sizeof* an integer type data is 2 that of float is 4, that of double is 8, that of char is1.

p

RELATIONAL AND LOGICAL OPERATORS

< (less than), <= (less than or equal to), > (greater than), >= (greater than or equal to), = = (equal to) and != (not equal to) are relational operators.

A logical expression is expression connected with a relational operator. For example ‘b*b – 4*a*c< 0 is a logical expression. Its value is either true or false.

```
int i, j, k ;
i=2;
j=3 ;
k=i+j ;
```

k>4 has the value true k<=3 has the value false.

LOGICAL OERATORS

The relational operators work with arbitrary numbers and generate true/false values. You can also combine true/false values by using the *Boolean operators*, which take true/false values as operands and compute new true/false values. The three Boolean operators are:

```
        &&      and
        ||      or
        !       not (takes one operand; ``unary'')
The && (``and'') operator takes two true/false values and produces
a true (1) result if both operands are true (that is, if the left-
hand side is true and the right-hand side is true). The || (``or'')
operator takes two true/false values and produces a true (1) result
```

```
if either operand is true. The ! (``not'') operator takes a single
true/false value and negates it, turning false to true and true to
false (0 to 1 and nonzero to 0).
```

&& (and) and || (or) are logical operators which are used to connect logical expressions. Where as ! (not) is unary operator, acts on a single logical expression.

For example,
```
1. (a<5) && (a>-2)
2. (a<=3) || (b>2)
```
In the first example if a= -3 or a=6 the logical expression returns true.

ASSIGNMENT OPERATORS

These operators are used for assigning a value of expression to another identifier.
=, + =, - = , * =, /= and %= are assignment operators.

a = b+c results in storing the value of b+c in 'a'.
a += 5 results in increasing the value of a by 5
a /= 3 results in storing the value a/3 in a and it is equivalent a=a/3

Note : **1**. if a floating point number is assigned to a integer type data variable, the value will be
truncated.

Example :
```
float a=5.36;
int b;
b=a
```
It results in storing 5 to b.

Similarly if an integer value is a assigned to a float type like float x=3 the value of x
stored is 3.0.

CONDITIONAL OPERATOR

The operator ?: is the conditional operator. It is used as
variable 1 = expression 1 ? expression 2 : expression 3.
Here expression 1 is a logical expression and expression 2 and expression 3 are expressions having numerical values. If expression 1 is true, value of expression 2 is assigned to variable 1 and otherwise expression3 is assigned.

Example :
```
int a,b,c,d,e
a=3;b=5;c=8;
d=(a<b)? a : b;
e=(b>c) ? b : c;
```

Then d=3 and e=8

LIBRARY FUNCTIONS

They are built in programs readily available with the C compiler. These function perform certain operations or calculations. Some of these functions return values when they are accessed and some carry out certain operations like input, output.a library functions accessed in a used written program by referring its name with values assigned to necessary arguments.
Some of these library functions are :
abs(i), ceil(d), cos(d), cosh(d), exp(d), fabs(d),floor(d), getchar( ), log(d), pow(d,d'),
printf( ), putchar(c), rand( ), sin(d), sqrt(d), scanf( ), tan(d), toascii(c), toupper(c), tolower(c).

Note : the arguments i, c, d are respectively integer, char and double type.

Example:

```
#include<math.h>
#include<stdio.h>
#<conio.h>
main( )
{
float x, s;
printf(" \n input the values of x :");
scanf("%f ", &x);
s=sqrt(x);
printf("\n the square root is %f ",s);
}
```

Note that C language is case sensitive, which means 'a' and 'A' are different. Before the main program there are statements begin with # symbol. They are called preprocessor statements. Within the main program " float r, a;" is a declaration statement. 'include' is a preprocessor statement. The syntax is #include<file name>. it is to tell the compiler looking for library functions, which are used in the program, included in the file, file name (like stdio.h, conio.h, math.h, etc…).