Expression and Operators

- The symbols which are used to perform logical and mathematical operations in a C program are called C operators.
- These C operators join individual constants and variables to form expressions.
- Operators, functions, constants and variables are combined together to form expressions.
- Consider the expression $A + B \times 5$.
- Where, $+$, $\times$ are operators, $A$, $B$ are variables, $5$ is constant and $A + B \times 5$ is an expression.

Types of C operators

- C language offers many types of operators.
- They are,
  - Arithmetic operators.
  - Assignment operators.
  - Relational operators.
  - Logical operators.
  - Bit wise operators.
  - Conditional operators (ternary operators).
  - Increment/decrement operators.
  - Special operators
Arithmetic Operators in C

Arithmetic operators are used to perform mathematical calculations like addition, subtraction, multiplication, division and modulus in C programs.

<table>
<thead>
<tr>
<th>S.no</th>
<th>Arithmetic Operators</th>
<th>Operation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>Addition</td>
<td>A+B</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>Subtraction</td>
<td>A-B</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>Multiplication</td>
<td>A*B</td>
</tr>
<tr>
<td>4</td>
<td>/</td>
<td>Division</td>
<td>A/B</td>
</tr>
<tr>
<td>5</td>
<td>%</td>
<td>Modulus</td>
<td>A%B</td>
</tr>
</tbody>
</table>

**Example program for C arithmetic operators**

- In this example program, two values "40" and "20" are used to perform arithmetic operations such as addition, subtraction, multiplication, division, modulus and output is displayed for each operation.

```c
#include <stdio.h>
int main()
{
    int a=40, b=20, add, sub, mul, div, mod;
    add = a+b;
    sub = a-b;
    mul = a*b;
    div = a/b;
    mod = a%b;
    printf("Addition of a, b is: %d\n", add);
    printf("Subtraction of a, b is: %d\n", sub);
    printf("Multiplication of a, b is: %d\n", mul);
    printf("Division of a, b is: %d\n", div);
    printf("Modulus of a, b is: %d\n", mod);
}
```

**Output:**

Addition of a, b is: 60  
Subtraction of a, b is: 20  
Multiplication of a, b is: 800  
Division of a, b is: 2  
Modulus of a, b is: 0
Assignment operators

- In C programs, values for the variables are assigned using assignment operators.
- For example, if the value "10" is to be assigned for the variable "sum", it can be assigned as "sum = 10;".
- Other assignment operators in C language are given below.

<table>
<thead>
<tr>
<th>Operators</th>
<th>Example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple assignment operator</td>
<td>=</td>
<td>sum=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 is assigned to variable sum</td>
</tr>
<tr>
<td></td>
<td>+=</td>
<td>sum+=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is same as sum = sum+10</td>
</tr>
<tr>
<td></td>
<td>-=</td>
<td>sum-=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is same as sum = sum-10</td>
</tr>
<tr>
<td>Compound assignment operators</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*=</td>
<td>sum*=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is same as sum = sum*10</td>
</tr>
<tr>
<td></td>
<td>/=</td>
<td>sum/=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is same as sum = sum/10</td>
</tr>
<tr>
<td></td>
<td>%=</td>
<td>sum%=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is same as sum = sum%10</td>
</tr>
<tr>
<td></td>
<td>&amp;=</td>
<td>sum&amp;=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is same as sum = sum&amp;10</td>
</tr>
<tr>
<td></td>
<td>^=</td>
<td>sum^=10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is same as sum = sum^10</td>
</tr>
</tbody>
</table>

Example program for C assignment operators

- In this program, values from 0 – 9 are summed up and total "45" is displayed as output.
- Assignment operators such as "*=" and "+=" are used in this program to assign the values and to sum up the values.

```c
#include <stdio.h>
int main()
{
    int Total=0,i;
    for(i=0;i<10;i++)
    {
        Total+=i; // This is same as Total = Total+i
    }
    printf("Total = %d", Total);
}
```

Output:

Total = 45
Relational operators

- Relational operators are used to find the relation between two variables, i.e., to compare the values of two variables in a C program.

<table>
<thead>
<tr>
<th>S.no</th>
<th>Operators</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&gt;</td>
<td>x &gt; y</td>
<td>x is greater than y</td>
</tr>
<tr>
<td>2</td>
<td>&lt;</td>
<td>x &lt; y</td>
<td>x is less than y</td>
</tr>
<tr>
<td>3</td>
<td>&gt;=</td>
<td>x &gt;= y</td>
<td>x is greater than or equal to y</td>
</tr>
<tr>
<td>4</td>
<td>&lt;=</td>
<td>x &lt;= y</td>
<td>x is less than or equal to y</td>
</tr>
<tr>
<td>5</td>
<td>==</td>
<td>x == y</td>
<td>x is equal to y</td>
</tr>
<tr>
<td>6</td>
<td>!=</td>
<td>x != y</td>
<td>x is not equal to y</td>
</tr>
</tbody>
</table>

Example program for relational operators in C

- In this program, relational operator (==) is used to compare 2 values whether they are equal are not.
- If both values are equal, output is displayed as "values are equal".
- Else, output is displayed as "values are not equal".
- Note: double equal sign (==) should be used to compare 2 values.
- We should not single equal sign (=).

```c
#include <stdio.h>
int main()
{
    int m=40, n=20;
    if(m == n)
    {
        printf("m and n are equal");
    }
    else
    {
        printf("m and n are not equal");
    }
}
```

Output:

m and n are not equal