Operators in C Programming

Lothe Savita A.

To study….

- Arithmetic operators
- Assignment operators
- Relational operators
- Logical operators
- Bit wise operators
- Conditional operators (ternary operators)
- Increment/decrement operators
- Special operators.

Introduction

- 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.

Types of C Operators

Types of Operators	Description
Arithmetic Operators	Perform mathematical calculations like addition, subtraction, multiplication, division and modulus
Assignment Operators	Assign the values for the variables in C programs.
Relational operators	Compare the value of two variables.
Logical operators	Perform logical operations on the given two variables.
Bit wise operators	Perform bit operations on given two variables.
Conditional (ternary) operators	Conditional operators return one value if condition is true and returns another value is condition is false.
Increment/decrement operators	Either increase or decrease the value of the variable by one.
Special operators	&, *, sizeof( ) and ternary operators.

ARITHMETIC OPERATORS IN C

Arithmetic Operators/Operation	Example
+ (Addition)	A+B
– (Subtraction)	A-B
* (multiplication)	A*B
/ (Division)	A/B
% (Modulus)	A%B

Example showing use of Arithmetic Operator

```
#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);
   return 0;
}
```

ASSIGNMENT OPERATORS IN C

Operators	Example/Description
=	test = 10; 10 is assigned to variable test
+=	test += 10; This is same as test = test + 10
-=	test -= 10; This is same as test = test – 10
*=	test *= 10; This is same as test = test * 10
/=	test /= 10; This is same as test = test / 10
%=	test %= 10; This is same as test = test % 10
&=	test&=10; This is same as test = test & 10
^=	test ^= 10; This is same as test = test ^ 10

Example showing use of Assignment Operator

```
# include <stdio.h>

int main()
{
int Total=0,i;
for(i=0;i<10;i++)
{
Total+=i; // This is same as Total = Toatal+i
}
printf("Total = %d", Total);
return 0;
}
```

RELATIONAL OPERATORS IN C

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

Operators	Example/Description
>	x > y (x is greater than y)
<	x < y (x is less than y)
>=	x >= y (x is greater than or equal to y)
<=	x <= y (x is less than or equal to y)
==	x == y (x is equal to y)
!=	x != y (x is not equal to y)

Example showing use of Relational Operator

```
#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");
   }
return 0;
}
```

LOGICAL OPERATORS IN C

Operators	Example/Description
&& (logical AND)	(a>5)&&(b<5) It returns true when both conditions are true
|| (logical OR)	(a>=10)||(b>=10) It returns true when at-least one of the condition is true
! (logical NOT)	!((a>5)&&(b<5)) It reverses the state of the operand "((a>5) && (b<5))" If "((a>5) && (b<5))" is true, logical NOT operator makes it false

Example showing use of Logical Operator

```
#include <stdio.h>
int main()
{
   int m=40,n=20;
   int a=20,p=30;
   if (m>n && m !=0)
   {       printf("&& Operator : Both conditions are true\n");    }
   if (a>p || p!=20)
   {       printf("|| Operator : Only one condition is true\n");     }
   if (!(m>n && m !=0))
   {      printf("! Operator : Both conditions are true\n");   }
   else
   {      printf("! Operator : Both conditions are true. " \
      "But, status is inverted as false\n");
   }
return 0;
}
```

BIT WISE OPERATORS IN C

- These operators are used to perform bit operations. Decimal values are converted into binary values which are the sequence of bits and bit wise operators work on these bits. Following are bitwise operator

1) & Bitwise AND
2) | Bitwise OR
3) ~ Bitwise NOT
4) ^ XOR
5) << Left Shift
6) >> Right Shift

Program showing Bitwise Operator in C

```
#include <stdio.h>
int main()
{
   int m = 40,n = 80,AND_opr,OR_opr,XOR_opr,NOT_opr ;
   AND_opr = (m&n);
   OR_opr = (m|n);
   NOT_opr = (~m);
   XOR_opr = (m^n);
   printf("AND_opr value = %d\n",AND_opr );
   printf("OR_opr value = %d\n",OR_opr );
   printf("NOT_opr value = %d\n",NOT_opr );
   printf("XOR_opr value = %d\n",XOR_opr );
   printf("left_shift value = %d\n", m << 1);
   printf("right_shift value = %d\n", m >> 1);
}
```

CONDITIONAL OR TERNARY OPERATORS IN C

- Conditional operators return one value if condition is true and returns another value is condition is false.
- This operator is also called as ternary operator.

Syntax : (Condition? true_value: false_value);

Example : (A > 100 ? 0 : 1);

Program for use of Ternary Operator

```
#include <stdio.h>
int main()
{
  int x=1, y ;
  y = ( x ==1 ? 2 : 0 ) ;
  printf("x value is %d\n", x);
  printf("y value is %d", y);
  return 0;
}
```

Increment / Decrement Operators

- Increment operators are used to increase the value of the variable by one and decrement operators are used to decrease the value of the variable by one in C programs.

Increment operator

++var_name; (or) var_name++;

Decrement operator

- -var_name; (or) var_name – -;

Example for increment operators

```
#include <stdio.h>
int main()
{
    int i=1;
    while(i<10)
    {
        printf("%d ",i);
        i++;
    }
return 0;
}
```

Example for Decrement operators

```
#include <stdio.h>
int main()
{
    int i=1;
    while(i<10)
    {
        printf("%d ",i);
        i--;
    }
return 0;
}
```

Thank You