Operators and Precedences

Course: BTech in CSE
Course Name: Programming for Problem Solving
Course Code:
Semester: II
Session: 2019-20

Mr. Joynath Mishra
Assistant Professor (Guest)
Department of Computer Science and Information Technology

Mahatma Gandhi Central University
Bihar, INDIA

April 28, 2020

Outline

Objectives

Objectives

- Study on operators and their precedences
- Steps to solve arithmetic expressions on C compiler

Introduction[1],[2],[3]

- Mathematical formula calculation are performed with variety of operators in computer language.
- Operators are used to formulate any mathematical expression and calculate its result
- Arithmetic operators require unary or binary operands in nature.
- Exprsions are mathematical functions that solve any equation.
- One expression may contain different data type, implicit type casting, utilize returning value from any function or program.

Operator % allows *int* values only, no *float* is applicable.

Table 1: Arithmetic Operators in C Language

Operator	Symbol	Operation	Syntax
Multiplication	*	x times y	$z = x * y$
Division	/	x divided by y	$z = x/y$
Remainder	%	remainder of x divided by y	$z = x\%y$
Addition	+	addition	z= $x + y$
Substraction	-	sustraction	$z = x - y$

Example

```
1  #include<stdio.h>
2  int main()
3  {
4      int x=5, y=3,z;
5
6      z = x+y;   printf("\nSum = %d", z);
7      z = x/y;   printf("\nQuotient = %d", z);
8      z = x%y;   printf("\nRemainder = %d", z);
9
10     return 0;
11 }
```

Output

Sum = 8
Quotient = 1
Remainder = 2

TRUE condition returns 1 and FALSE condition returns 0.

Table 2: Relational Operators in C Language

Operator	Symbol	Operation	Syntax
Greater than	>	greater than	x > y;
Greater than or equal	>=	Greater than or equal	x >= y
Less than	<	less than	x<y
Less than or equal	<=	less than or equal	x <= y
Equal	==	equal checking	x == y
Not Equal	!=	not equal checking	x != y

Example

```
1  #include<stdio.h>
2  int main()
3  {    int x=5, y=3, z;
4       z = x>y;     printf("\n1st Value = %d", z);
5       z = x>=y;    printf("\n2nd Value = %d", z);
6       z = x<y;     printf("\n3rd Value = %d", z);
7       z = x<=y;    printf("\n4th Value = %d", z);
8       z = x==y;    printf("\n5th Value = %d", z);
9       z = x!=y;    printf("\n6th Value = %d", z);
10      return 0;
11 }
```

Output

```
1st Value = 1
2nd Value = 1
3rd Value = 0
4th Value = 0
5th Value = 0
6th Value = 1
```

Table 3: Logical Operators in C Language

Operator	Symbol	Operation	Syntax
AND	&&	logical AND of two condition	z = x==y && x<y
OR	||	logical OR of two condition	z = x==y || x<y
NOT	!	logical NOT	

Example

```
1  #include<stdio.h>
2  int main()
3  {
4      int x=3, y=4, z;
5      z = 3 & 4;      printf("\nValue of z: %d", z);
6      z = 3 | 4;      printf("\nValue of z: %d", z);
7      z = ~3;         printf("\nValue of z: %d", z);
8      printf("\n");
9      z = 3 && 4;     printf("\nValue of z: %d", z);
10     z = x && y;     printf("\nValue of z: %d", z);
11     z = x || y;     printf("\nValue of z: %d", z);
12     z = !x;         printf("\nValue of z: %d", z);
13     return 0;
14 }
```

Output

```
Value of z: 0
Value of z: 7
Value of z: -4

Value of z: 1
Value of z: 1
Value of z: 1
Value of z: 0
```

Operator % allows *int* values only, no *float* is applicable.

Table 4: Assignment Operators in C Language

Operator	Symbol	Operation	Syntax
z = x	=	value assignment	z = x
x = x-y	-=	substruction and assignment	x -= y
x = x+y	+=	addition and assignment	x += y
x = x*y	*=	multiplication and assignment	x *= y
x = x/y	/=	quotient and assignment	x /= y
x = x%y	%=	remainder and assignment	x %= y

Example

```
1  #include<stdio.h>
2  int main()
3  {    int x=5, y=3;
4       x -= y;    printf("\n1st Value = %d", x);
5       x += y;    printf("\n2nd Value = %d", x);
6       x *= y;    printf("\n3rd Value = %d", x);
7       x /= y;    printf("\n4th Value = %d", x);
8       x %= y;    printf("\n5th Value = %d", x);
9       return 0;
10 }
```

Output

```
1st Value = 2
2nd Value = 5
3rd Value = 15
4th Value = 5
5th Value = 2
```

Table 5: Pointer Operator in C Language

Operator	Symbol	Operation	Syntax
Pointer	*	denote address of a variable	p = &x;
Pointer indicator	->	denote variable of pointer structure	

Example

```
1  #include<stdio.h>
2  int main()
3  {
4      int x = 500;
5      int *p, **q;
6      p = &x;
7      q = &p;
8      printf("\nAddress of x : %p", p);
9      printf("\nAddress of x : %p", &x);
10     printf("\nAddress of p : %p", q);
11     printf("\nAddress of p : %p", &p);
12     printf("\nValue of x : %d", *p);
13     printf("\nValue of x : %d", **q);
14
15
16     return 0;
17 }
```

Output

```
Address of x : 0x7ffcfbd7e6b4
Address of x : 0x7ffcfbd7e6b4
Address of p : 0x7ffcfbd7e6b8
Address of p : 0x7ffcfbd7e6b8
Value of x : 500
Value of x : 500
```

Table 6: Arithmetic Operators in C Language

Operator	Symbol	Operation	Syntax
AND	&	binary AND operation	z = x & y
OR	|	binary OR operation	z = x | y
XOR		ˆ binary XOR operation	z = x ŷ
1's complement	~	binary complement	z = ~x
Right shift	»	binary digit right shift	z = x » y
Left shift	«	binary digit left shift	z = x « y

Example

```
1  #include<stdio.h>
2  int main()
3  {    int x=5, y=3,z;
4
5       z = x & y;   printf("\nAND Operation = %d",z);
6       z = x | y;  printf("\nOR Operation = %d",z);
7       z = x ^ y;   printf("\nXOR Operation = %d",z);
8       z = ~x;      printf("\nComplement Operation = %d",z);
9       z = x >> y;  printf("\nRight shift Operation = %d",z);
10      z = x << y;  printf("\n Left shift Operation = %d",z);
11
12      return 0;
13 }
```

Output

```
AND Operation = 1
OR Operation = 7
XOR Operation = 6
Complement Operation = -6
Right shift Operation = 0
Left shift Operation = 40
```

Table 7: Special Operator in C Language

Operator	Symbol	Operation	Syntax
Comma	,	separate variables	int x,y;
sizeof	sizeof()	return size of variable	sizeof(x)

Example

```
#include<stdio.h>
int main()
{
    int a, x, y, z;
     x = printf("\nDear MGCUB Students\t");
     y = scanf("%d%d%d%d", &a,&a,&a,&a);
     z = sizeof(scanf("%d%d%d%d", &a,&a,&a,&a));
     printf("\nValue of x: %d", x);
     printf("\nValue of y: %d", y);
     printf("\nValue of z: %d", z);
     return 0;
}
```

Output

```
Dear MGCUB Students 1 2 3 4

Value of x: 21
Value of y: 4
Value of z: 4
```

Table 8: Conditional Operator in C Language

Operator	Symbol	Operation	Syntax
Ternary	?:	condition check	z = x>y ? x : y

Example

```
1 #include<stdio.h>
2 int main()
3 {
4     int a;
5     a = 4>8?5:6;
6      printf("Value = %d", ++a);
7      return 0;
8
9 }
```

Output

Value = 7

Precedences of Arithmetic Operators

Table 9: Operators Precedences in C Language

Operator	Associativity	Precedence
++ or −	Right to Left	↓
* / %	Left to Right	
+-	Left to Right	Lowest

Precedences of Relational and Logical Operators

Table 10: Operators Precedences in C Language

Operator	Associativity	Precedence
!	Right to Left	
> >= < <=	Left to Right	↓
== !=	Left to Right	
&&	Left to Right	
||	Left to Right	Lowest

Example

Example 01

```c
#include<stdio.h>
int main()
{
    double x;
    x = 2 * 5 / 4 + 3 / 4 + 8 - 6 + 5 / 8;
    printf("Value of x: %lf", x);
    return 0;
}
```

Output

Value of x: 4.000000

Example 02

```c
#include<stdio.h>
int main()
{
    int x;
    x = 2 - 3 + 5 * 2 / 8 % 3;
    printf("Value of x: %d", x);
    return 0;
}
```

Output

Value of x: 0

Example (Contd...)

Example 03

```c
#include<stdio.h>
int main()
{
    int x;
    x = 2 * 3 + 5 - 10 / 3;
    printf("Value of x: %d", x);
    return 0;
}
```

Output

Value of x: 8

Example 04

```c
#include<stdio.h>
int main()
{
    int a=1, b=2, c=3, d=4, e=5, f=6;
    int result;
    result = a * -b + d % e - f;
    printf("Value of result: %d", result);
    return 0;
}
```

Output

Value of result: -4

Example (Contd...)

Example 05

```c
#include<stdio.h>
int main()
{
    int x;
    x = 12 > (7 + 5);
    printf("Value of x: %d", x);
   return 0;
}
```

Output

Value of x: 0

Example 06

```c
#include<stdio.h>
int main()
{
    int x;
    x = (4 > 3) && (100 != 200);
    printf("Value of x: %d", x);
    return 0;
}
```

Output

Value of x: 1

Exercise

- Solve arithmetic expression for x.
  - $x = 2 + 4 - 6 * 8/2/2$ Solution: x = -6
  - $x = (5\&8)||(!4)$ Solution: x = 0
  - $x = 4 < 9 > 4 + 5$ Solution: x = 0
  - $x = 5 - 3\%5 + 9 * 4/3$ Solution: x = 14
- Solve arithmetic expression for x where a=1, b=2, c=3, d=4, e=5, f=6
  - $x = a\&b$ Solution: x = 0
  - $x = c - d + f * a * a$ Solution: x = 5
  - $x = d > c||a < b$ Solution: x = 1
  - $x = ++d + c\%a$ Solution: x = 5

References I


S. Jain, *Programming and Problem Solving through C Language Design*. BPB Publications, 2003.


K. Yashwant, "Let us c," *Array and pointers, 7th edition, BPB publication*, 1999.


——, *Test Your C Skills*. BPB publication, 2005.

Get in touch via...

+91 9046174189

jaynath4025@gmail.com

Thank You...