Tutorial Link https://codequotient.com/tutorials/C - More Operators/5a1d9a1d52795c1b16c0ac0f

TUTORIAL

C - More Operators

Chapter

Topics

There are other useful operators in C language as below:

Assignment

In C, (=) equals sign is a assignment operator. It takes two operands. First on left side whose value will get the new value assigned (known as lvalue), and second on right side whose value will be assigned to lvalue (known as rvalue). For example,

```
int a=56;   // a (lvalue) will be assigned 56 (rvalue).
char c='A';   // c (lvalue) will be assigned 'A' (rvalue).
```

Lvalue is an object, which referes to a memory location, it should be modifiable otherwise you will get an error. For example,

```
5 = 6;      // 6 is correct rvalue, but 5 is not an lvalue, as 5
is a constant value not a location.
```

You can assign multiple variables in a single statement as below: -

```
int a, b, c, d;
a = b = c = d = 10;      // it will assign 10 to all these
valriables.
```

Assignment operator start assignment from right side and returns the rvalue, hence, first d will be assigned 10, and it returns 10, so next c will be assigned 10, and so on.

The & and * Operators

These are used with pointer management.

sizeof

sizeof is a unary compile-time operator that returns the length, in bytes, of the variable or parenthesized type specifier that it precedes. For example,

```c
#include <stdio.h>

int main()
{
  char c;
  int a;
  float b;
  printf("size of char variable is = %ld\n", sizeof(c));
  printf("size of int variable is = %ld\n", sizeof(a));
  printf("size of float variable is = %ld\n", sizeof(b));
  printf("size of double variable is = %ld\n", sizeof(double));
  printf("size of short int variable is = %ld\n", sizeof(short int));
  printf("size of long int variable is = %ld\n", sizeof(long int));
  printf("size of long double variable is = %ld\n", sizeof(long double));

  return 0;
}

```

Although the size varies from platform to platform, so the output will vary from computer to computer.

The parameter to sizeof() function will not get executed, i.e. if we execute the below line: -

```
printf("%d", sizeof(printf("HELLO")));
```

Then HELLO will not print, instead printf() functions returns an integer so sizeof() function will return the size of an integer variable. So only 4 will be printed.

Comma

It is used to sequencing of operations. If multiple operations are to be performed, we can sequence them using comma operator. It is quite tricky sometimes to use this operator. For example,

```
#include <stdio.h>

int main()
{
  int a=5, b=2, c;
  c = (a++, a-b);
  printf("a = %d b = %d c = %d\n", a, b, c);

  return 0;
}

```

Here c = (a++, a-b); will be evaluated as below: -

first a++ will be performed, which makes a = 6, now a-b will return 4 which is assigned to c.

Dot (.) and Arrow (->)

These are used to access the individual members of structures and unions.

[] and ()

[] are used with arrays, and () are used in an expression to override the implicit order of evaluation and also to call a function.