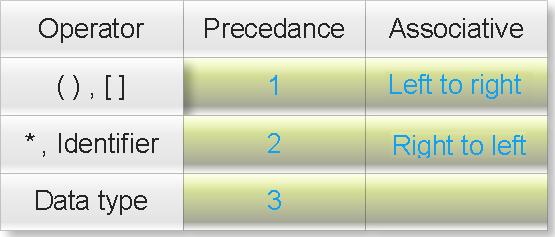
**How to read complex pointers in C Programming?**

**Rule 1.** Assign the priority to the pointer declaration considering precedence and associative according to following table.

[](https://3.bp.blogspot.com/_uIwyaTjqYYw/Sv1zkDcfr1I/AAAAAAAABKI/C-SYjll0r9A/s1600-h/precedance.jpeg)

Where

**():** This operator behaves as bracket operator or function operator.

**[]:** This operator behaves as array subscription operator.

**\***: This operator behaves as pointer operator not as multiplication operator.

**Identifier**: It is not an operator but it is name of pointer variable. You will always find the first priority will be assigned to the name of pointer.

**Data type**: It is also not an operator. Data types also includes modifier (like signed int, long double etc.)

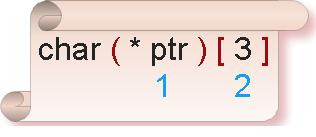
You will understand it better by examples:

**(1) How to read following pointer?**

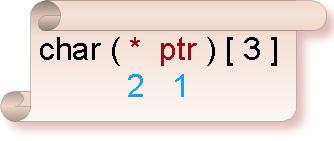
char (\* ptr)[3]

Answer:

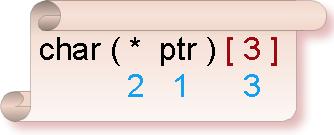
**Step 1:** () and [] enjoys equal precedence. So rule of associative will decide the priority. Its associative is left to right So first priority goes to ().



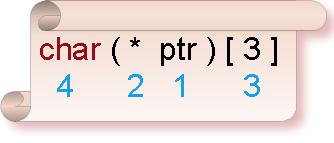
**Step 2:** Inside the bracket \* and ptr enjoy equal precedence. From rule of associative (right to left) first priority goes to ptr and second priority goes to \*.



**Step3:** Assign third priority to [].



**Step4:** Since data type enjoys least priority so assign fourth priority to char.



Now read it following manner:

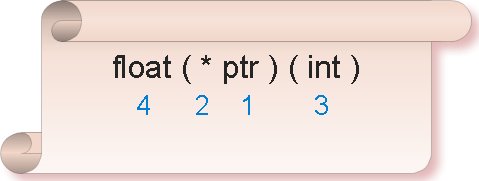
**ptr** is **pointer** to such one dimensional **array** of size three which content **char** type data.

(2) How to read following pointer?

float (\* ptr)(int)

Answer:

Assign the priority considering precedence and associative.



Now read it following manner:

**ptr** is **pointer** to such **function** whose parameter is int type data and return type is **float** type data.

**Rule 2:**Assign the priority of each function parameter separately and read it also separately.

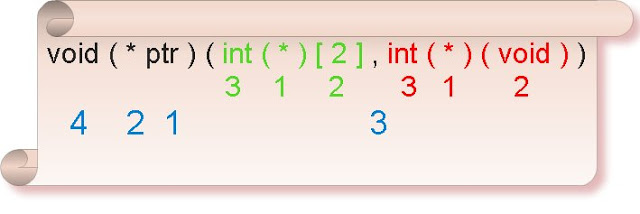
Understand it through following example.

(3) How to read following pointer?

void (\*ptr)(int (\*)[2],int (\*) void))

Answer:

Assign the priority considering rule of precedence and associative.



Now read it following manner:

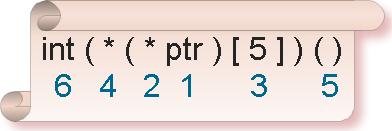
**ptr** is **pointer** to such **function** which first parameter is **pointer** to one dimensional **array** of size two which content **int** type data and second parameter is **pointer** to such **function** which parameter is void and return type is **int** data type and return type is **void**.

(4) How to read following pointer?

int ( \* ( \* ptr ) [ 5 ] ) ( )

Answer:

Assign the priority considering rule of precedence and associative.



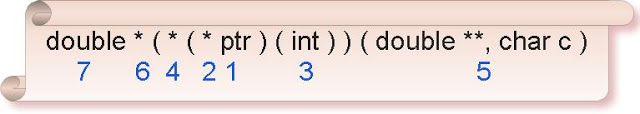
Now read it following manner:

**ptr** is **pointer** to such **array** of size five which content are **pointer** to such **function** which parameter is void and return type is **int** type data.

(5) How to read following pointer?

double\*(\*(\*ptr)(int))(double \*\*,char c)

Answer:



Assign the priority considering rule of precedence and associative.

Now read it following manner:

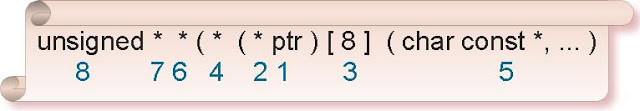
**ptr** is **pointer** to **function** which parameter is int type data and return type is **pointer** to **function** which first parameter is pointer to pointer of double data type and second parameter is char type data type and return type is **pointer** to **double** data type.

(6) How to read following pointer?

unsigned \*\*(\*(\*ptr)[8](char const \*, ...)

Answer:

Assign the priority considering rule of precedence and associative.



Now read it following manner:

**ptr** is **pointer** to **array** of size eight and content of array is **pointer** to **function** which first parameter is pointer to character constant and second parameter is variable number of arguments and return type is **pointer** to **pointer** of **unsigned** int data type.