What is a Function?

It is block of code which takes input, processes it and produce output.

Functions run only when it is called.

Types of functions

- User-defined
    1. add()
    2. display()
    3. show()
- Pre-defined
    1. strcpy()
    2. strcmp()

    etc

User-defined function syntax :-

```
return_type function_name(parameters) {
  //block of code
}
```

Recursion

When a function call itself the function is known as recursive function and the mechanism is known as recursion.

Syntax-

```
return_type function_name(parameter) {
     if(base_condition) {
         //code
     } else {
         Function_name(parameter);
     }
}
```

Call by value

In call by value the actual value of variable cannot be changed, if we change the value of function parameter it is only changed for current function.

Call By address

It is used when we want to modify a variable inside a function and want that modification persist outside the function.

Syntax

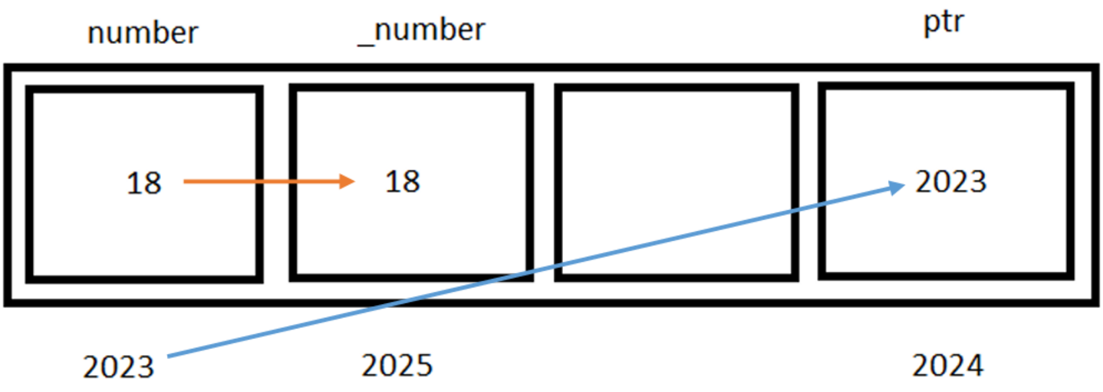
```
return_type function_name(*p) {
   //modification
 }
```

Call By Reference

It is alias to an existing variable. When we declare a reference, we create a new name for an existing variable, and any changes done in reference is treated as if it were the original variable.

Pointer

It is a variable that holds the memory address of another variable.



A pointer variable points to a data type (like int) of the same type, and is created with the * operator.

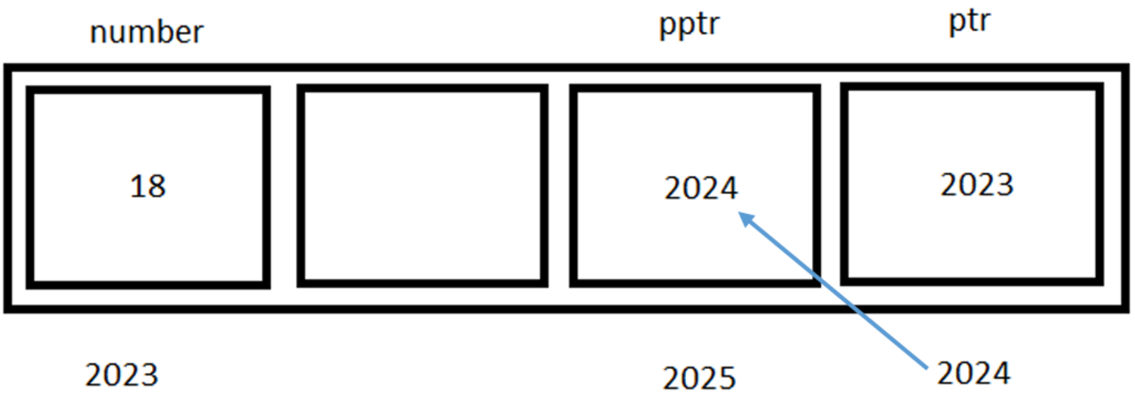
Syntax

```
int number = 18; // * = value at address operator
int *ptr = &number; // & = address of operator
int _number = *ptr; // storing the value in another variable

cout<<number; //a = 18
cout<<ptr;        //ptr = 2023
cout<<*ptr;       //*ptr = 18
cout<<&number;        //&number = 2023
```

Pointer to Pointer

A variable that stores the memory address of another pointer.


number
pptr
ptr
18
2024
2023
2023
2025
2024