Chapter 10 / User-Defined Functions

Category of functions: Generally a function contain function name, ~~function~~ arguments, return type and also definition of this function. Based on these ~~categories function~~ function may belong on of the following categories.

Function with no arguments and ~~return type~~ no return values: When a function has no arguments, it does not receive any data from the calling function. Similarly, when it does not return a value, the calling function doesn't receive any data from the called function.

Example,

```
void function () {
    ......
}
main ()
{
    ......
    function ();
    ......
}
```

Function with Arguments but no return value: When a function has no return value. But there present arguments, or function parameters.

Example:

```
void function(a1,a2,a3.....an); formal arguments
main()
{
.....
function(f1,f2,f3.....fn); actual arguments
.....
}
```

a array of character

Function with arguments and a return value:

In this type of functions are receives data from calling function and called function return a value to the calling function.

Example:

```
return_type function(f.){
.....
return(e);
}
main()
{
.....
variable = function(a);
.....
}
```

*Here, return_type means a data_type, which data type a function needs to return.

2023/2/17 22:41

Function with no arguments but a returns a value

This type of function is important when we need to design a function, that takes no value from the calling function but returns a value to calling function.

Example:

```
return_type function() {
    ---
    return(e);
}
main()
{
    ---
    variable = function();
    ---
}
```

Here, return types means a data type, which data type need to return the called function.

Function with that return multiple values,

When a function have to return multiple values, that time we design this function with pointer to variable. Using pointer we can be return multiple values.