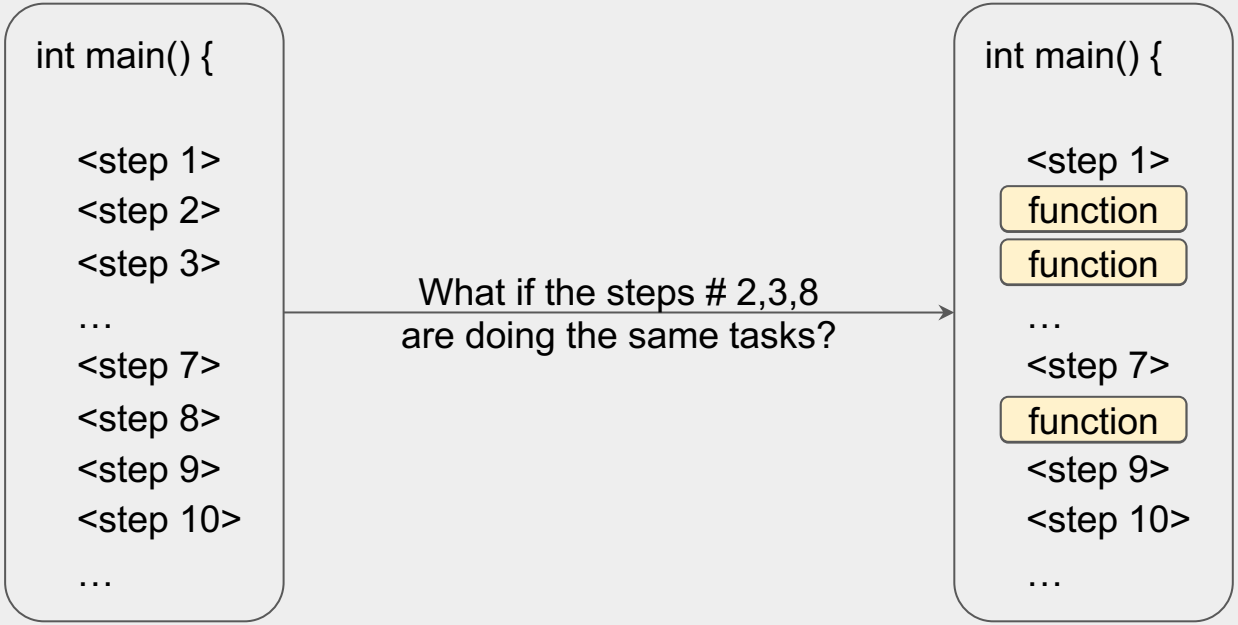
Function

Introduction to Computers and Programming
Week6 TA Course
2023/10/17

Why do we need FUNCTION?

Let’s take a look at this situation-



Why do we need FUNCTION?

Let's take a look at this example-

```
int main() {

    printf("Hi\n");
    printf("I am a programmer.\n");
    printf("----------------------\n");
    …
    printf("I like coding.\n");
    printf("----------------------\n");
    printf("Just kidding…\n");
    printf("----------------------\n");
    …
```



```
void sep_line() {
    print("----------------------\n");
}

int main() {

    printf("Hi\n");
    printf("I am a programmer.\n");
    sep_line();
    …
    printf("I like coding.\n");
    sep_line();
    printf("Just kidding…\n");
    sep_line();
    …
```

Another Example

Calculating the average score of each student

Student Num	Chinese	English	Math	Society	Science
0	80	85	89	78	90
1	70	82	95	67	93
2	88	74	85	73	98

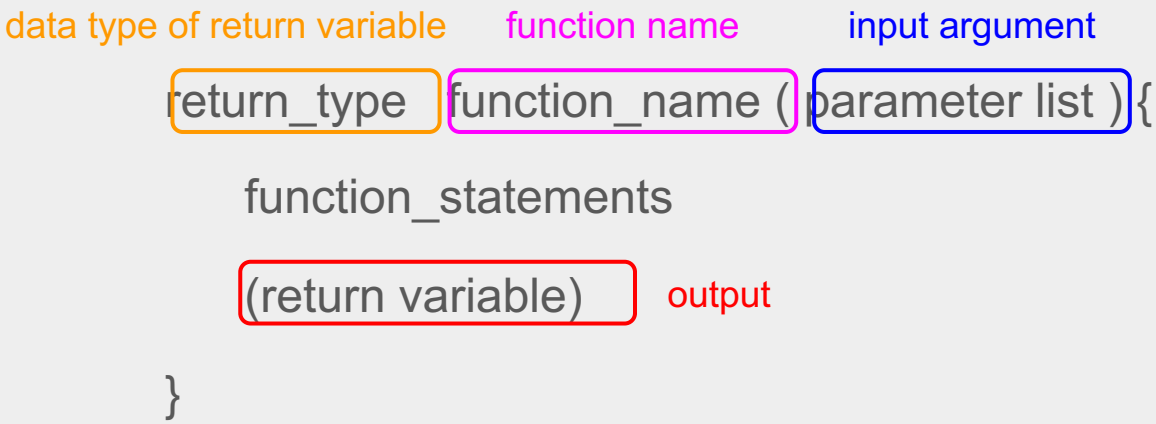
⋮

```
float cal_avg(int array[]) {
    // do some stuff
    return average_score;
}
```

```
int main() {

    int scores[30][5];
    for (int i=0; i<30; i++) {
avg = cal_avg(scores[i]);
printf("Student %d avg: %f\n");
    }
    …
```

Function Define Format



Function Define Format

data type of return variable　function name　input argument

```
float find_the_mean ( int a, int b ) {
    float avg= (float)(a+b)/2;
    return avg;
}
```

output

How To Use?

```
int x=5, y=2;
float ans;
ans = find_the_mean(x, y);
```

Function Define Format

data type of return variable　　function name　　input argument

```
void print_sth ( char str[ ] ) {
    printf("%s", str);
    return nothing
}
```

~~output~~

How To Use?

```
char str_in[10] = "abcd";
print_sth(str_in);
```

2 Ways to Declare a Function

```
float  find_the_mean (int a, int b) {
    float avg= (float)(a+b)/2;
    return avg;
}
int  main () {
    int x=5, y=2;
    float ans;
    ans = find_the_mean(x, y);
    printf("The mean of %d and %d is %f", x, y, ans);
    return 0;
}
```

Declare the function first, otherwise your program will not be acquainted to it when you call the function.

Or, you can declare the **function prototype** first.

```
float  find_the_mean (int, int);
int  main () {
    int x=5, y=2;
    float ans;
    ans = find_the_mean(x, y);
    printf("The mean of %d and %d is %f", x, y, ans);
    return 0;
}
float  find_the_mean (int a, int b) {
    float avg= (float)(a+b)/2;
    return avg;
}
```

More Detail:

- Pass by value:

  Any changes made to the parameter within the function do not affect the original value outside of the function.

  Example. In C, integers, floats, characters and so on...

- Pass by reference:

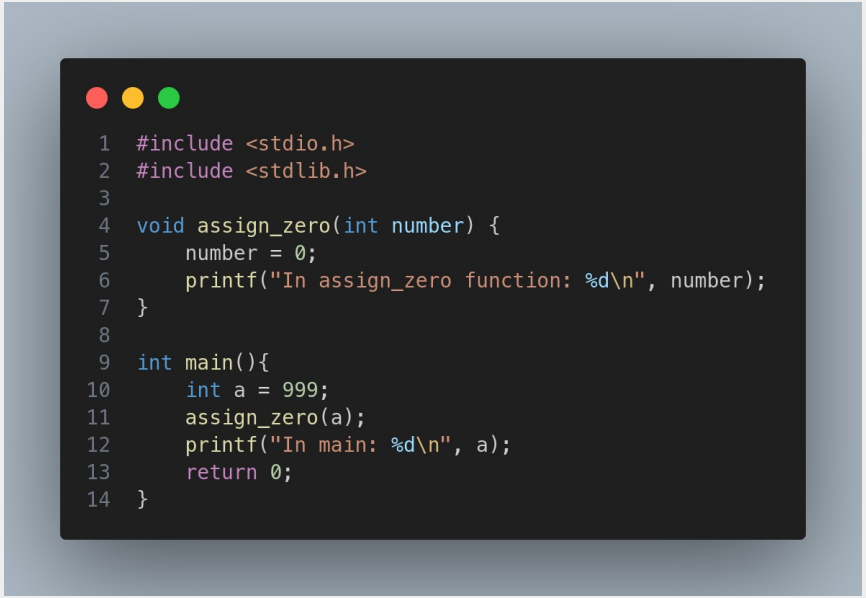
  Any changes made to the parameter within the function directly affect the original data outside of the function.

  Example. In C, it doesn’t have pass by reference, but we can pass pointer to achieve it.

  In C, passing the entire array is essentially passing a pointer to the head of the array.

More Detail:

- Pass by value:
- Any changes made to the parameter within the function do not affect the original value outside of the function.



```c
#include <stdio.h>
#include <stdlib.h>

void assign_zero(int number) {
    number = 0;
    printf("In assign_zero function: %d\n", number);
}

int main(){
    int a = 999;
    assign_zero(a);
    printf("In main: %d\n", a);
    return 0;
}
```

Output:
In assign_zero function: 0
In main: 999

More Detail:

- Pass by reference:
- Any changes made to the parameter within the function directly affect the original data outside of the function.

```c
#include <stdio.h>
#include <stdlib.h>

void assign_zero(int number_list[]) {
    number_list[0] = 0;
    printf("In assign_zero function ([0]): %d\n", number_list[0]);
}

int main(){
    int arr[3] = {9, 9, 9};
    assign_zero(arr);
    printf("In main ([0]): %d\n", arr[0]);
    return 0;
}
```

Output:
In assign_zero function ([0]): 0
In main ([0]): 0

Exercises!

Exercise 1

- Scan 2 values, ***x*** and ***y***, which are integers.
- Find the smaller of 2 given numbers by the function **min()**.
- Print the input arguments and the output answer.
  - e.g. “The min of 12 and 31 is 12.\n”
- Edit the TODO and declare your function.
- Example
  - Input: 20 15
  - Output: The min of 20 and 15 is 15.

Use this as template

```
int  main () {
    /* TODO */
     ans = min(x, y);
    /* TODO (print the ans) */
     system(“pause”);
     return 0;
}
```

Exercise 2

- Scan 3 values, n, r, and a_1, which are all natural numbers.
- Calculate the geometric sequence with common ratio r,
  - started from a_1, and
  - there should be n terms in the sequence.
  - the outcome should be stored in ‘geometric_arr’.
- Print the sequence from a_1 to a_n.
- n <= 10, r <= 8, a <= 10
- Declare your function and call it.
- Example
  - Input: 5 3 1
  - Output: 1 3 9 27 81

Use this as template

```
/*Define 'createGeometric' function*/

int main(){
                int geometric_arr[10] = {-1};
                int n, r, a;
                scanf("%d %d %d", &n, &r, &a);
               createGeometric(); // According your function definition,
input some parameters into this function

                for (int i = 0; i < n; ++i){
                                printf("%d\n", geometric_arr[i]);
                }

                return 0;
}
```