

C Function Parameters

```
Topics : <u>C</u>
Written on <u>April 13, 2023</u>
```

Function parameters are the inputs to a function that allow it to perform a specific task. In C, there are two ways to pass function parameters: by value and by reference.

Passing parameters by value means that the function receives a copy of the original variable, and any modifications made to the parameter inside the function do not affect the original variable. Here's an example code snippet that shows how to pass parameters by value in C:

```
#include <stdio.h>
void swap(int num1, int num2) {
    int temp = num1;
    num1 = num2;
    num2 = temp;
}
int main() {
    int a = 10, b = 20;
    printf("Before swap: a = %d, b = %d\n", a, b);
    swap(a, b);
    printf("After swap: a = %d, b = %d\n", a, b);
    return 0;
}
```

Let's go over this code line by line:

- We first include the standard input/output library stdio.h.
- We define a function called swap that takes two integer arguments num1 and num2 and swaps their values using a temporary variable.
- In the main() function, we declare two integer variables called a and b and initialize them with the values 10 and 20, respectively.
- We use the printf() function to print the values of a and b before and after calling the swap function.
- Finally, we return 0 to indicate successful execution of the program.

When you run this program, it will print the following output:

Before swap: a = 10, b = 20After swap: a = 10, b = 20

Note that the values of a and b do not change after calling the swap function, because the function received copies of the original variables and any modifications made to the copies do not affect the original variables.

Passing parameters by reference means that the function receives the memory address of the original variable, and any modifications made to the parameter inside the function affect the original variable. Here's an example code snippet that shows how to pass parameters by reference in C using pointers:

```
#include <stdio.h>
void swap(int *ptr1, int *ptr2) {
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}
int main() {
    int a = 10, b = 20;
    printf("Before swap: a = %d, b = %d n", a, b);
    swap(&a, &b);
    printf("After swap: a = %d, b = %d\n", a, b);
    return 0;
```

}

Let's go over this code line by line:

- We first include the standard input/output library stdio.h.
- We define a function called swap that takes two integer pointers ptrl and ptr2 as arguments and swaps their values using a temporary variable and pointer dereferencing.
- In the main() function, we declare two integer variables called a and b and initialize them with the values 10 and 20, respectively.
- We use the printf() function to print the values of a and b before and after calling the swap function.
- We pass the memory addresses of a and b to the swap function using the & operator.

© Copyright Aryatechno. All Rights Reserved. Written tutorials and materials by Aryatechno