**//Call by Value Example - Swapping 2 numbers using Call by Value**

#include <stdio.h>

void swap(int, int);

int main()

{

int x, y;

printf("Enter the value of x and y\n");

scanf("%d%d",&x,&y);

printf("Before Swapping\nx = %d\ny = %d\n", x, y);

swap(x, y);

return 0;

}

void swap(int a, int b)

{

int temp;

temp = b;

b = a;

a = temp;

printf("Values of x and y after swapping are %d %d\n",a,b);

}

The **call by reference** method of passing arguments to a function copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call. It means the changes made to the parameter affect the passed argument.

To pass a value by reference, argument pointers are passed to the functions just like any other value. So accordingly you need to declare the function parameters as pointer types as in the following function **swap()**, which exchanges the values of the two integer variables pointed to, by their arguments.

/\* function definition to swap the values \*/

void swap(int \*x, int \*y) {

int temp;

temp = \*x; /\* save the value at address x \*/

\*x = \*y; /\* put y into x \*/

\*y = temp; /\* put temp into y \*/

return;

}

Let us now call the function **swap()** by passing values by reference as in the following example −

#include <stdio.h>

/\* function declaration \*/

void swap(int \*x, int \*y);

int main () {

/\* local variable definition \*/

int a = 100;

int b = 200;

printf("Before swap, value of a : %d\n", a );

printf("Before swap, value of b : %d\n", b );

/\* calling a function to swap the values.

\* &a indicates pointer to a ie. address of variable a and

\* &b indicates pointer to b ie. address of variable b.

\*/

swap(&a, &b);

printf("After swap, value of a : %d\n", a );

printf("After swap, value of b : %d\n", b );

return 0;

}