Course Name: O Level (2nd Sem B1 Batch) **Subject:** C Language
Topic: Function in C (Call by reference) **Date:** 15-April-2020

❑ Call by reference in C

- In call by reference, the address of the variable is passed into the function call as the actual parameter.
- The value of the actual parameters can be modified by changing the formal parameters since the address of the actual parameters is passed.
- In call by reference, the memory allocation is similar for both formal parameters and actual parameters. All the operations in the function are performed on the value stored at the address of the actual parameters, and the modified value gets stored at the same address.

Difference between Call by Value and Call by Reference

CALL BY VALUE	CALL BY REFERENCE
While calling a function, we pass values of variables to it. Such functions are known as "Call By Values".	While calling a function, instead of passing the values of variables, we pass address of variables (location of variables) to the function known as "Call By References".
In this method, the value of each variable in calling function is copied into corresponding temporary variables of the called function.	In this method, the addresses of actual variables in the calling function are copied into the temporary variables of the called function.
With this method, the changes made to the temporary variables in the called function have no effect on the values of actual variables in the calling function.	With this method, using addresses we would have an access to the actual variables and hence we would be able to manipulate them.

Example 1: Write a user defined function to swap the value of two variables.

```
#include <stdio.h>
#include <conio.h>
void swap(int*, int*);                  /* function declaration */
void main ()
{
  int a=10,b=20;
  clrscr();
  printf("Before swap value of a : %d\n",a);
  printf("Before swap value of b : %d\n",b);
  swap(&a,&b);                          //function call and passing the address of a & b

  printf("After swap, value of a : %d\n", a );
  printf("After swap, value of b : %d\n", b );
  getch();
}
void swap(int *x, int *y)
{
   int t;
   t=*x;                                //copy the value at address x into t
   *x=*y;                               //copy the value at address y into value at address x
   *y=t;                                //copy the value of t into value at address y
}
```

Output :

```
Before swap value of a : 10
Before swap value of b : 20
After swap, value of a : 20
After swap, value of b : 10
```

'*' called **'value at address'** operator. It gives the value stored at a particular address.
The **'value at address'** operator is also called **'indirection'** operator.

Example 2: **Write user defined function to input any number and print the factorial value of the number.**

```
#include<stdio.h>
#include<conio.h>
int fact(int*);
void main()
{
int x,y;
clrscr();
printf("Enter any number");
scanf("%d",&x);
y=fact(&x);
printf("Factorial value of%d is%d",x,y);
getch();
}
int fact(int *z)
{
int i;
for(i=1;*z>1;(*z)--)
i=i*(*z);
return (i);
}
```

Example 3: **Write user defined function to input any number. Check the number is positive or negative.**

```
#include<stdio.h>
#include<conio.h>
int posi(int *);
void main()
{
int x,y;
clrscr();
printf("Enter any number\n");
scanf("%d",&x);
y=posi(&x);
y==1 ? printf("Number is positive") : printf("Number is negative");
getch();
}
int posi(int *z)
{
if(*z>=0)
return 1;
else
return 0;
}
```

Example 4: Write user defined function to input any two numbers and check which one is greater.

```
#include<stdio.h>
#include<conio.h>
int greater(int*,int*);
void main()
{
int x,y,z;
clrscr();
printf("Enter any two number\n");
scanf("%d%d",&x,&y);
z=greater(&x,&y);
z==1 ? printf("x is greater number ") : printf("y is greater number");
getch();
}
int greater(int *a, int *b)
{
if(*a > *b)
return 1;
else
return 0;
}
```

Try Yourself:

1. Write user defined function to input any year and check the year is leap year or not leap year.
2. Write user defined function to input any number and check the number is positive or negative.
3. Write user defined function to input the selling and cost price and check the profit or loss.