SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 23CST101 PROBLEM SOLVING AND C PROGRAMMING

I YEAR/ V SEMESTER

UNIT – IV FUNCTIONS AND POINTERS

CALL BY VALUE

Dr.B.Vinodhini

Associate Professor

Department of Computer Science and Engineering

FUNCTIONS

Two ways in which we can pass arguments to Functions

Call by Value

- Value of arguments are passed to called function
- Operation is done in formal Parameter
- Changes made are local to that function
- Once Came out of function the changes made get vanish

Call by Reference

Call by Reference rather than passing value address(reference) are passed .Function operates on addresses rather than values .Formal arguments points to actual arguments changes made are permanent

Call By Value

Values of actual parameters will be copied to formal parameters and these two different parameters store values in different locations

```
int x = 10, y = 20;
fun(x, y);
printf("x = %d, y = %d", x, y);
```

```
int fun(int x, int y)
{
    x = 20;
    y = 10;
}
```

Output: x = 10, y = 20

Call by Value Example: Swapping the values of the two variables

```c
#include <stdio.h>
void swap(int , int); //prototype of the function
int main()
{
   int a = 10;
   int b = 20;
   printf("Before swapping the values in main a = %d, b = %d\n",a,b); // printing the value of a and b in main
   swap(a,b);
   printf("After swapping values in main a = %d, b = %d\n",a,b); // The value of actual parameters do not change by
}
void swap (int a, int b)
{
   int temp;
   temp = a;
   a=b;
   b=temp;
   printf("After swapping values in function a = %d, b = %d\n",a,b); // Formal parameters, a = 20, b = 10
}
```

Output

```
Before swapping the values in main a = 10, b = 20
After swapping values in function a = 20, b = 10
After swapping values in main a = 10, b = 20
```


Thank
you