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 REFERENCE

Dr.B.Vinodhini

Associate Professor

Department of Computer Science and Engineering

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
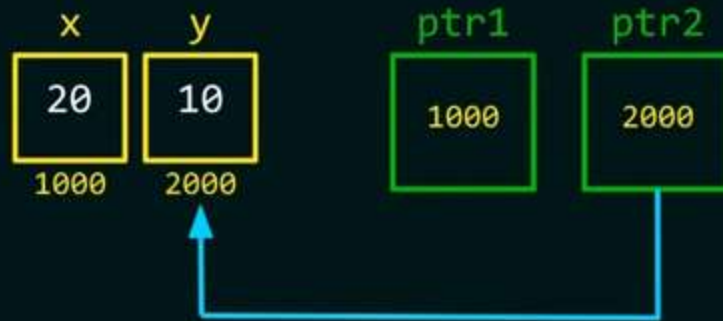
Here both actual and formal parameters refers to same memory location .Therefore any

changes made to the formal parameters will get reflected to actual parameters

Here instead of passing values we pass addresses

```
int x = 10, y = 20;              int fun(int *ptr1, int *ptr2)
fun(&x, &y);                     {
                                     *ptr1 = 20;
printf("x = %d, y = %d", x, y);      *ptr2 = 10;

                                 }

Output: x = 20, y = 10

        x       y           ptr1       ptr2

       20      10           1000       2000
     1000    2000
```

```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 values of actual parameters do change in call
}
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 = 20, b = 10
```