UNIVERSITY OF NIGERIA

SEMESTER EXAMINATIONS

SE/2018/2019

(Write on both sides of the Answer Sheet)

QUESTION No. _____ Exam No _____ Date _____

DO NOT WRITE ON THE MARGIN

Parameter Passing Technique

Actual Parameters: The parameters passed ~~by a~~ to a function.

Formal parameter: The parameters received by a function

```
add(m,n);                    int add(int a, int b)
                             {
                                return (a+b);
                             }
```

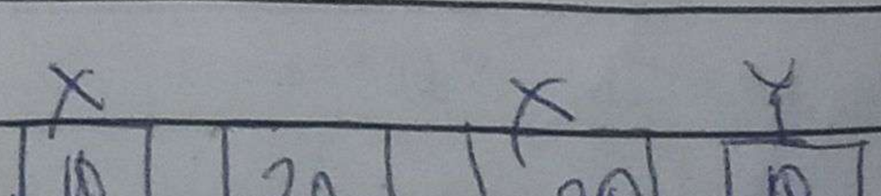
Actual parameters — Formal parameters

Call by value

Here 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;              int fun(int x, int y)
                             {
Fun(x,y);                       x=20;
Printf("x=%d, y=%d", x,y);      y=10;
                             }
```

Output = x=10, y=20



changing the value of x=20 & Y=10

* → D referene operator

DO NOT WRITE ON THE MARGIN

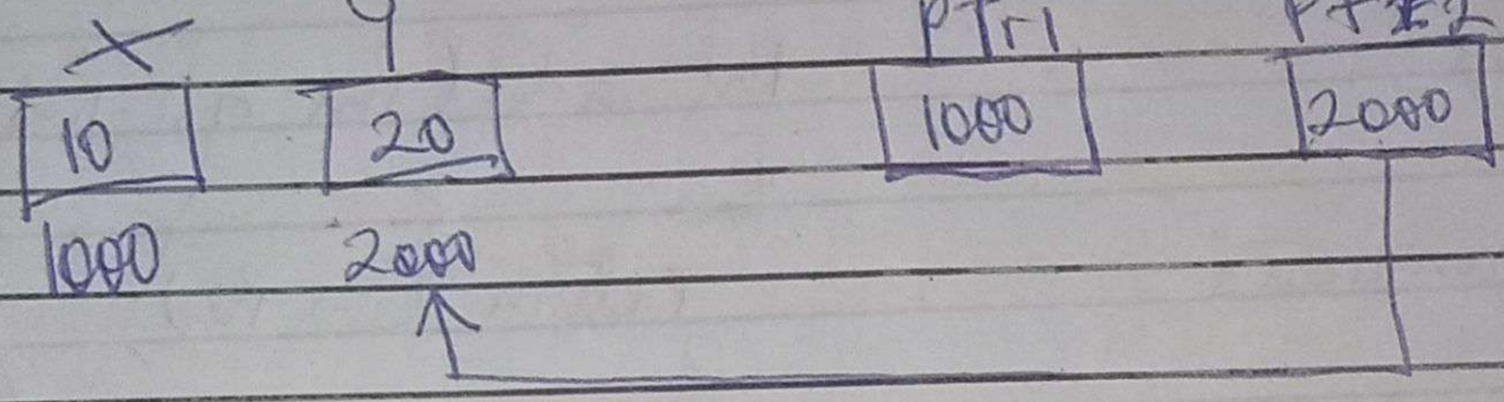
Call By Referenn

Here both actual anel formal parameters refers to same memory location. Therefor, any changes Made to the formal parameters will get reflected to the actual parameters. Here instead of passing values, we pass addresses.

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

Output: x = 20, y = 10

```
int fun(int* Ptr1, int* Ptr2)
{
    *Ptr1 = 20;
    *Ptr2 = 10;
}
```



Example 1 Using C++

```
#include <iostream>
// Function to swap two integers by value
void SwapByValue(int a, int b)
{
// Create a temporary variable to store
// the value of "a"
int temp = a;
// Assign the value of 'b' to "a"
a = b
// Assign the value of the temporary variable
// to "b"
b = temp;
}
```

UNIVERSITY OF NIGERIA

SEMESTER EXAMINATIONS

SE/2018/2019

(Write on both sides of the Answer Sheet)

QUESTION No. _____ Exam No _____ Date _____

DO NOT WRITE ON THE MARGIN

```
int main() {
    // Declare two integer variables, 'x' and 'y'
    // and initialize them
    int x = 5;
    int y = 10;
    // Print the values of 'x' and 'y' before
    // Swapping
    std::cout << "Before swapping: x = " << x << ",
        y = " << y << std::endl;
    // Call the SwapByValue function with
    // 'x' and 'y'
    SwapByValue(x, y);
    // Print the values of 'x' and 'y' after
    // swapping (which will remain unchanged)
    std::cout << "After swapping: x = " << x << ",
        y = " << y << std::endl;
    return 0;
}
```

Example 2: Passing parameter by Reference using C++

```
#include <iostream>
// Function to swap two integers by reference
void swapByReference(int &a, int &b)
{
    // We receive 'a' and 'b' by reference, meaning
    // we working directly with the original values
    // No need for temporal variable or return values
```

```
int temp = a; // store the value of 'a' in 'temp'
    a = b;    // Assign the value of 'b' to 'a'.
    b = temp  // ✓  ✓  ✓ 'temp' (which
              // was the original 'a') to 'b'
}
int main() {
    int x=5;
    int y = 10;
std::cout << "Before Swapping: x=" << x <<",
        y =" << y << std::endl;
// Call the ~~Swapping~~ SwapByReference function
// with 'x' and 'y' by reference.
    SwapByReference (x, Y);
    std::Cout << "After Swapping: x =" << x <<",
        y =" << y << std::endl;
    return 0;
}
```

1. We have a SwapByReference function that takes two integer references 'a' and 'b' as parameters. When parameters are passed by reference, the function works with the original values, allowing modifications to affect the caller's variable
2. In the main function, we declare two integer variable 'x' and 'y', and initialize them with values 5 and 10, respectively.
3. We Print the values of 'x' and 'y' before calling the SwapByReference function to see the original values.
4. We call the SwapByReference function with 'x' and 'y' as references. Inside

UNIVERSITY OF NIGERIA

SEMESTER EXAMINATIONS

SE/2018/2019

(Write on both sides of the Answer Sheet)

QUESTION No. _____ *Exam No* _____ *Date* _____

DO NOT WRITE ON THE MARGIN

The function, it swaps the values of 'a' and 'b' without using temporal variable because its working with the Original values.

5 After returning from the 'swapByReference' function, let print the values of 'x' and 'y' again. This time, you'll see that they have been swapped.

Using python programing language

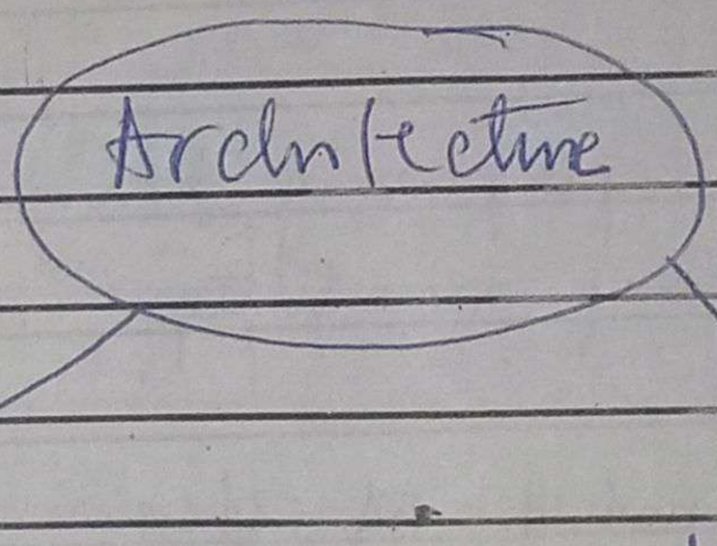
```
# Function to swap two integer
def Swap (a, b)
# We receive 'a' and 'b' as references, meaning we
  are working directly with the Original values
  temp = a # Store the value of 'a' in 'temp'.
    a = b # Assign the value of 'b' to 'a'.
    b = temp # Assign the value of 'temp' (which was the
  # Original 'a') to 'b'.
# Main Function
if __name__ == "__main__":
    X = 5
    Y = 10
    Print (f"Before Swapping: X = {X}, Y = {Y}")
# Call the swap function with 'x' and 'y'
    Swap (x, Y)
    Print (f"After Swapping: x = {X}, Y = {Y}")
    # In python we don't explicitly specify whether
```

DO NOT WRITE ON THE MARGIN

they are passed by value or reference. Python is dynamic language and everything is passed by Object reference.

Computer Architecture

This is the design of computers, including their instruction sets, hardware components and System Organization



$X = 2 + 3$

Instructional set Architecture (ISA)

LDI 02H MOV R1, 02H

ADI 03H or MOV R2, 03H or ADD X, 02+03H or Push

STA X ADD R1, R2 Push

STORE X, R1 ADD

Pop X