1. **Call by value**: The original variable will not be modified . We make separate copies of the original variable when passed as a parameter. Eg: in swap (a,b) the copies are passed in the function.
   1. Void main(){

{

a = 5, b = 3;

swap(a,b);

print(a,b);

}

Void swap(x,y)

{

X = x+y;

Y = x – y;

X = x-y;

}

* 1. The final values are as follow:
     1. a = 5, b = 3
     2. x = 3, y = 5
  2. Since x and y are not passed back, print(a,b) will print 5,3.

1. Call by reference: The original variable will get modified if the parameter variables are modified
   1. It is slower because we have to first get the address, then go to that address and then modify the value
   2. Each modification swap(x,y) will modify value of a,b
   3. Void main(){

{

a = 5, b = 3;

swap(a,b);

print(a,b);

}

Void swap(x,y)

{

X = x+y;

Y = x – y;

X = x-y;

}

* 1. The final value of a is 3 and b is 5.

1. Call by value-result:
   1. Similar to call by reference
   2. But only final result of swap(x,y) will modify a,b.
   3. Result may not necessarily equal to the result of call by reference if the method contains modification of global variable