

# Argument Passing in C

Call By Value vs Call by Reference

By

**Prof. Muhammad Iqbal Bhat**

Department of Higher Education  
Government Degree College Beerwah

# Topics:

1

Call by Value

2

Call by Reference

3

Examples

Prof. M. Iqbal Bhat (JKHED)

# Argument Passing in C



Call by value  
(default)

Call by  
reference



Prof. M. Iqbal Bhat (JNUED)

# Call by Value



When an argument is passed by value, a copy of the value is created and passed to the function.



Any changes made to the parameter inside the function do not affect the value of the original argument.



By default, arguments (other than arrays) are passed by value



Examples of types that are passed by value: integers, floats, characters, and structures.

# Call by Value: Example-1

```
int main() {  
    int num = 10;  
    printf("Before Function call: %d\n", num); // Output: 10  
    changeValue(num);  
    printf("After Function call: %d\n", num); // Output: 10  
    return 0;  
}  
void changeValue(int num) {  
    num = 5;  
}
```

# Call by Value: Example-2

```
// Cube a variable using pass-by-value.  
#include <stdio.h>
```

```
int cubeByValue(int n); // prototype
```

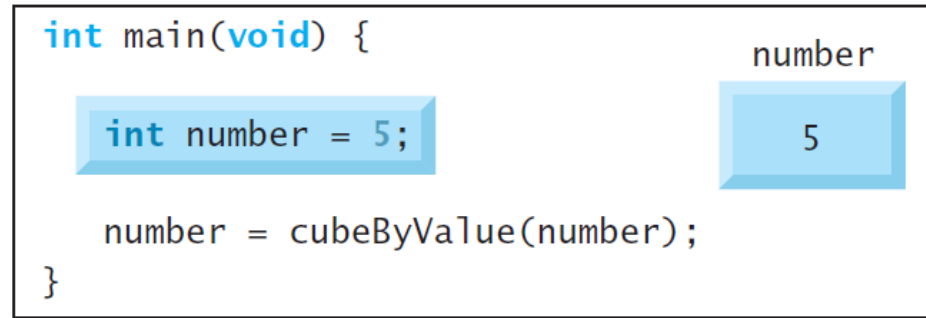
```
int main(void) {  
    int number = 5; // initialize number  
  
    printf("The original value of number is %d", number);  
    number = cubeByValue(number); // pass number by value to cubeByValue  
    printf("\nThe new value of number is %d\n", number);  
}
```

```
// calculate and return cube of integer argument
```

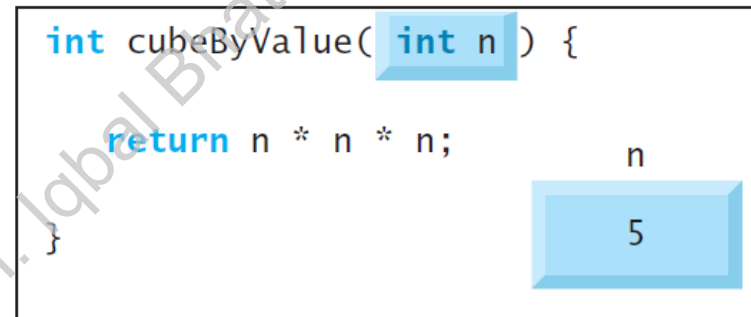
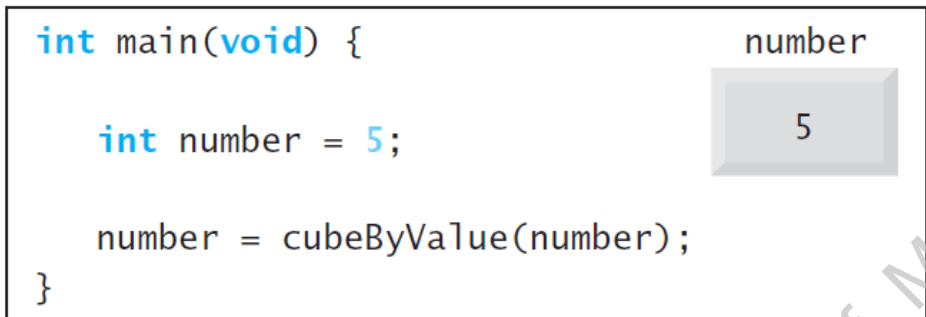
```
int cubeByValue(int n) {  
    return n * n * n; // cube local variable n and return result  
}
```

# Analysis of call by value

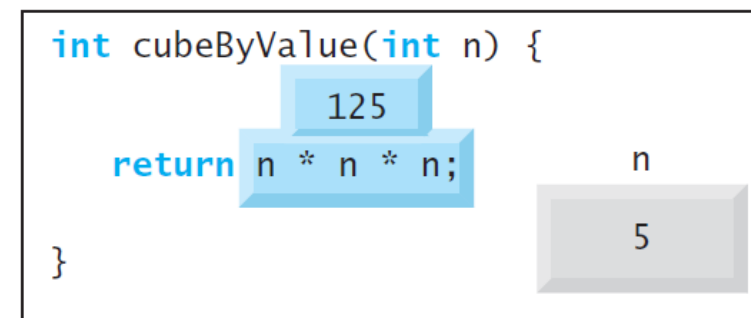
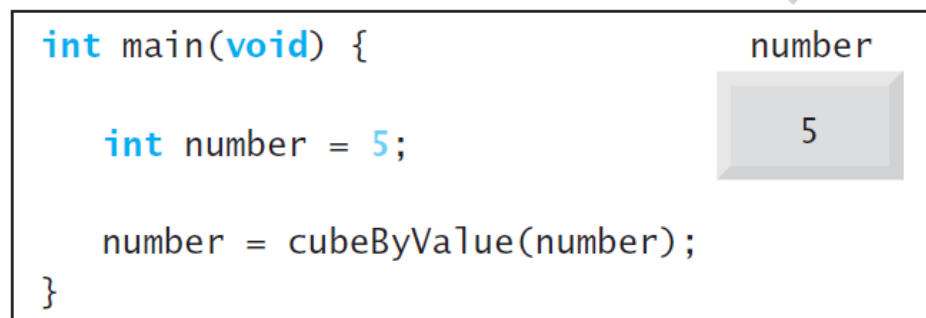
Step 1: Before main calls cubeByValue:



Step 2: After cubeByValue receives the call:

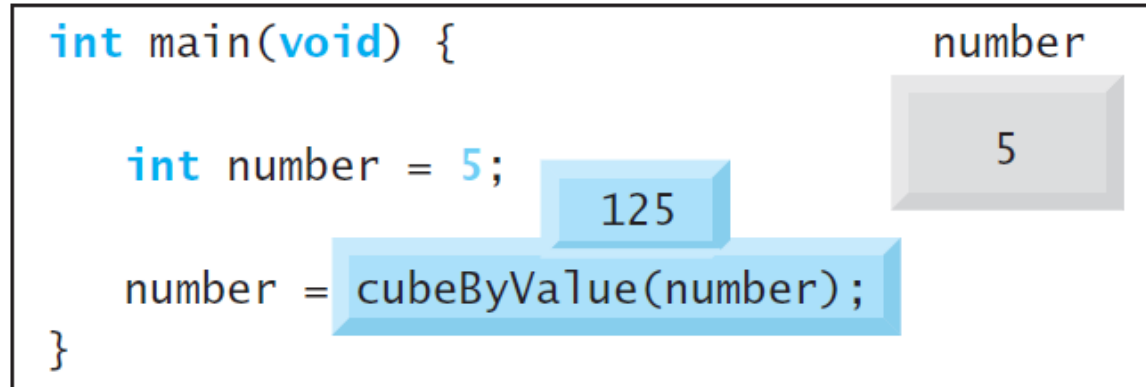


Step 3: After cubeByValue cubes parameter n and before cubeByValue returns to main:

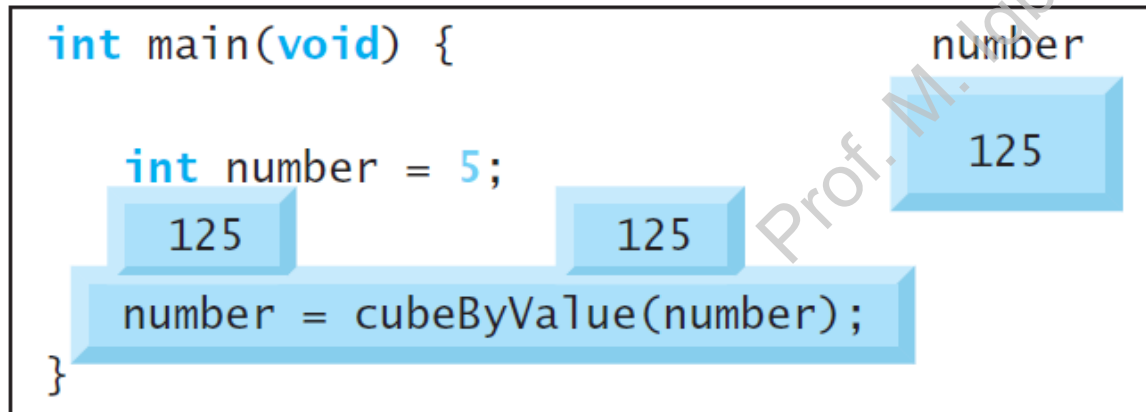


# Analysis of call by value

Step 4: After cubeByValue returns to main and before assigning the result to number:



Step 5: After main completes the assignment to number:





# Call by Reference

When an argument is passed by reference, the memory address of the argument is passed to the function.

Any changes made to the parameter inside the function affect the value of the original argument.

In C, we use pointers to pass arguments by reference.

When calling a function with arguments that should be modified in the caller, you use & to pass each variable's address.

Examples of types that are passed by reference: arrays and structures.

# Call by Reference: Example-1

```
int main() {
    int num = 10;
    printf("Before Function call: %d\n", num); // Output: 10
    changeValue(&num);
    printf("After Function call: %d\n", num); // Output: 5
    return 0;
}

void changeValue(int *num) {
    *num = 5;
}
```

# Call by Reference: Example-2

```
#include <stdio.h>
```

```
void cubeByReference(int *nPtr); // function prototype
```

```
int main(void) {
```

```
    int number = 5; // initialize number
```

```
    printf("The original value of number is %d", number);
```

```
    cubeByReference(&number); // pass address of number to cubeByReference
```

```
    printf("\nThe new value of number is %d\n", number);
```

```
}
```

```
// calculate cube of *nPtr, actually modifies number in main
```

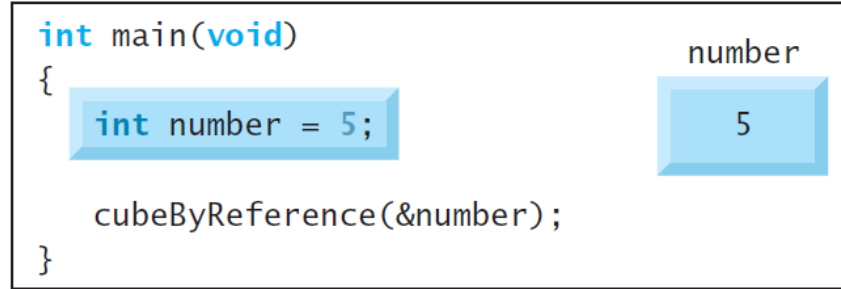
```
void cubeByReference(int *nPtr) {
```

```
    *nPtr = *nPtr * *nPtr * *nPtr; // cube *nPtr
```

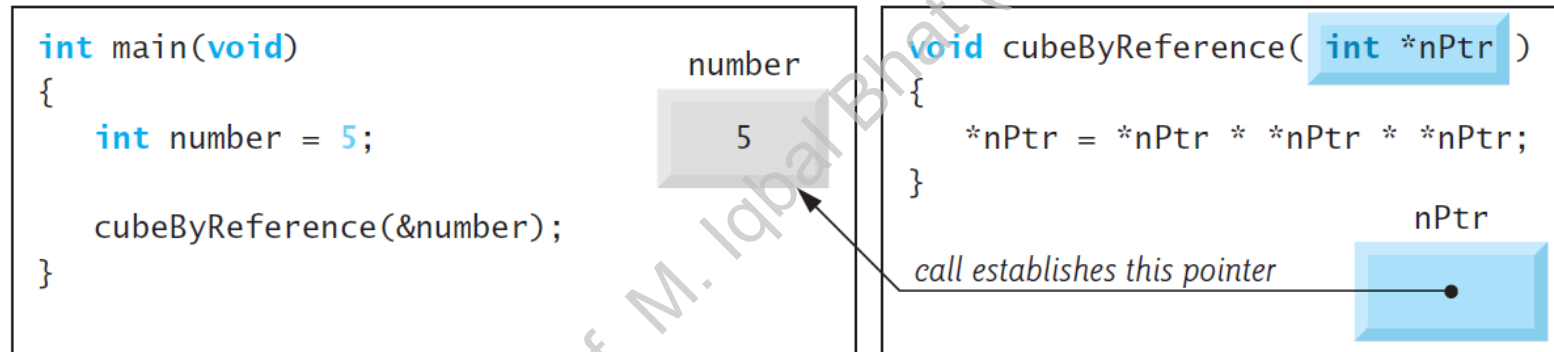
```
}
```

# Analysis of Call by Reference

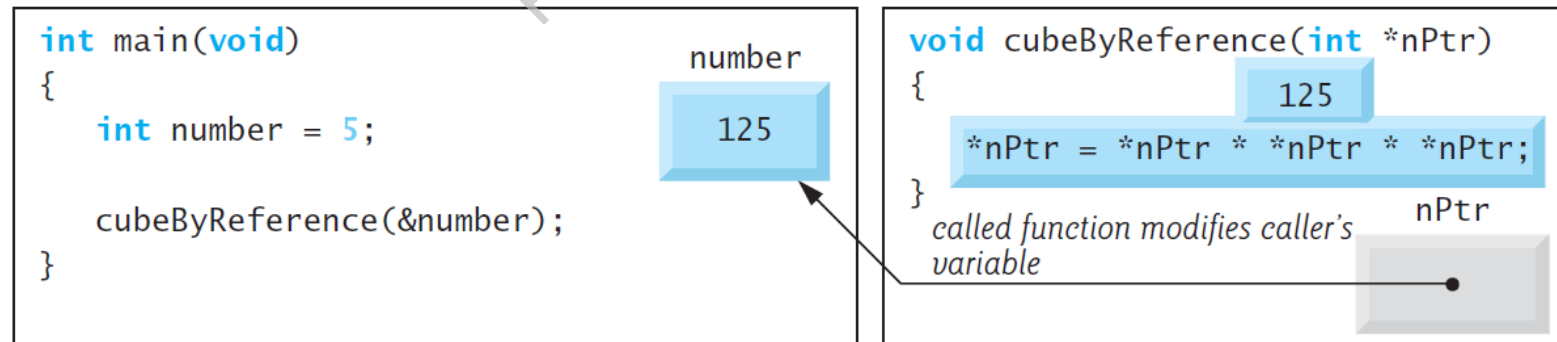
Step 1: Before main calls cubeByReference:



Step 2: After cubeByReference receives the call and before \*nPtr is cubed:



Step 3: After \*nPtr is cubed and before program control returns to main:

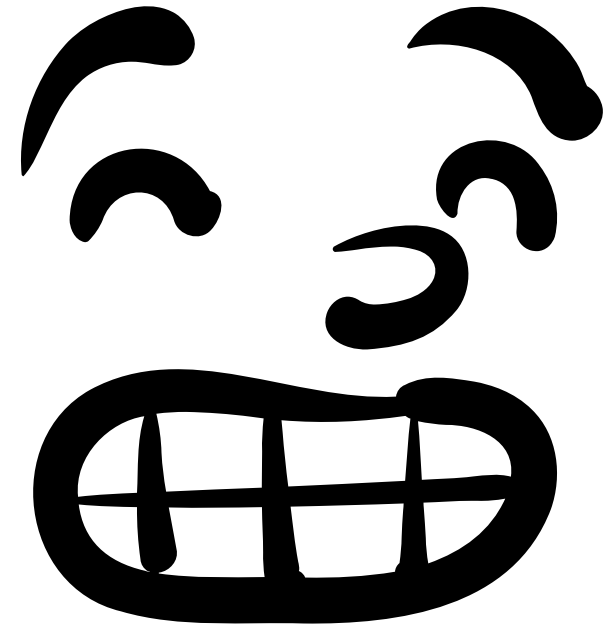


# Examples:

```
#include <stdio.h>

void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}

int main() {
    int x = 5, y = 10;
    printf("Before Swap: x=%d, y=%d\n", x, y);
    // Output: Before Swap: x=5, y=10
    swap(x, y);
    printf("After Swap: x=%d, y=%d\n", x, y);
    // Output: After Swap: x=5, y=10
    return 0;
}
```

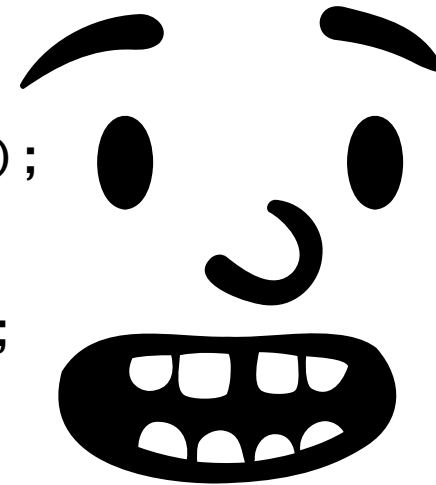


# Examples:

```
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int x = 5, y = 10;
    printf("Before Swap: x=%d, y=%d\n", x, y);
    // Output: Before Swap: x=5, y=10
    swap(&x, &y);
    printf("After Swap: x=%d, y=%d\n", x, y);
    // Output: After Swap: x=10, y=5
    return 0;
}
```





Prof. M. Iqbal Bhat (JKHED)

**Questions?**

