

3. Parameters and Arguments

3.1 Terminology

- **Parameters (Formal Parameters):** Variables in the function definition
- **Arguments (Actual Parameters):** Values passed when calling the function

```
// 'a' and 'b' are parameters
int add(int a, int b) {
    return a + b;
}

int main() {
    int x = 5, y = 3;
    // 'x' and 'y' are arguments
    int result = add(x, y);
    return 0;
}
```

3.2 Parameter Passing in C

C uses **pass by value** as its primary parameter passing mechanism. This means:

- A copy of the argument's value is made
- Changes to parameters inside the function don't affect the original variables
- The original variables remain unchanged

Example:

```
#include <stdio.h>

void modify_value(int num) {
    num = 100; // This only changes the local copy
    printf("Inside function: %d\n", num); // Output: 100
}

int main() {
```

```
int original = 50;
printf("Before function call: %d\n", original); // Output: 50

modify_value(original);

printf("After function call: %d\n", original); // Output: 50 (unchanged!)
return 0;
}
```

3.3 Python vs C Parameter Passing

Python:

```
def modify_list(lst):
    lst.append(4) # Modifies the original list

my_list = [1, 2, 3]
modify_list(my_list)
print(my_list) # Output: [1, 2, 3, 4]
```

C (Basic Types):

```
void modify_number(int num) {
    num = 100; // Only modifies the copy
}

int main() {
    int number = 50;
    modify_number(number);
    // number is still 50
    return 0;
}
```

3.4 Multiple Parameters

Functions can have multiple parameters of different types:

```
#include <stdio.h>

// Function to calculate compound interest
```

```
float calculate_compound_interest(float principal, float rate, int time, int n) {
    float amount = principal;
    float rate_per_period = rate / (100.0 * n);
    int total_periods = n * time;

    for (int i = 0; i < total_periods; i++) {
        amount *= (1 + rate_per_period);
    }

    return amount - principal; // Return only the interest
}

int main() {
    float principal = 1000.0;
    float annual_rate = 5.0;    // 5% per year
    int years = 2;
    int compounding_freq = 4;   // Quarterly

    float interest = calculate_compound_interest(principal, annual_rate, years,
compounding_freq);

    printf("Principal: $%.2f\n", principal);
    printf("Interest earned: $%.2f\n", interest);
    printf("Total amount: $%.2f\n", principal + interest);

    return 0;
}
```

Revision #1

Created 2025-09-06 10:16:42 UTC by DS

Updated 2025-09-06 10:17:16 UTC by DS