

Recursion

Pass by value

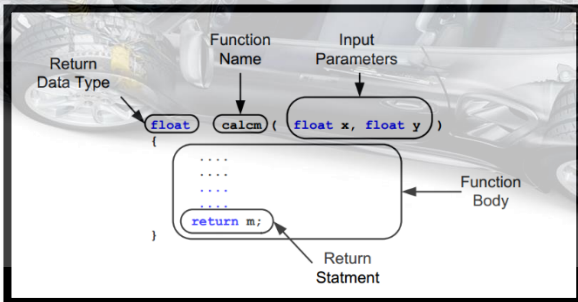
This method copies **the actual value** of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.

Pass by reference

This method copies **the address** of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call. This means that changes made to the parameter affect the argument.

Function Definition

Functions

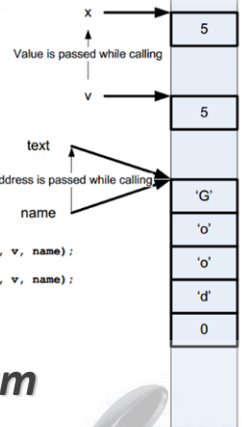


```

#include <stdio.h>

void tryToModify(int x, char text[])
{
    x++;
    text[0]--;
}

void main()
{
    int v = 5;
    char name[5] = "Good";
    printf("v = %d, name = %s\n", v, name);
    tryToModify(v, name);
    printf("v = %d, name = %s\n", v, name);
}
  
```



Memory layout of C-program

Prototype

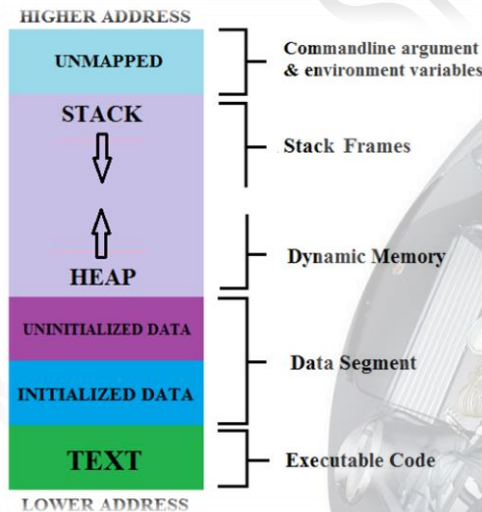
```

#include <stdio.h>

int addNumbers(int a, int b);

int main()
{
    ...
    sum = addNumbers(n1, n2);
    ...
}

int addNumbers(int a, int b)
{
    ...
}
  
```

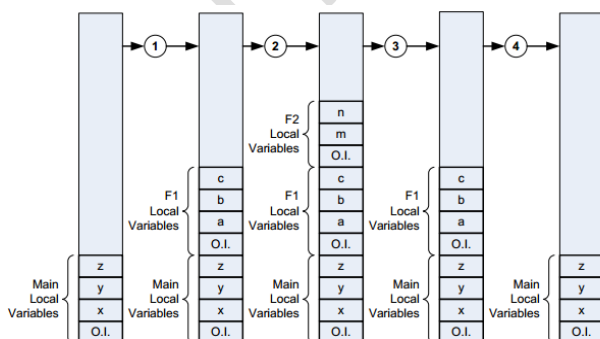


Local variables

Dynamic Allocation (malloc, ...)

Static And global

Stack Calling Mechanism



```

void main()
{
    int x = 3, y = 6, z;
    z = f1(x, y);
    printf("Square of (%d + %d) = %d\n", x, y, z);
}

int f1(int a, int b)
{
    int c;
    c = f2(a + b);
    return c;
}

int f2(int m)
{
    int n;
    n = m * m;
    return n;
}
  
```