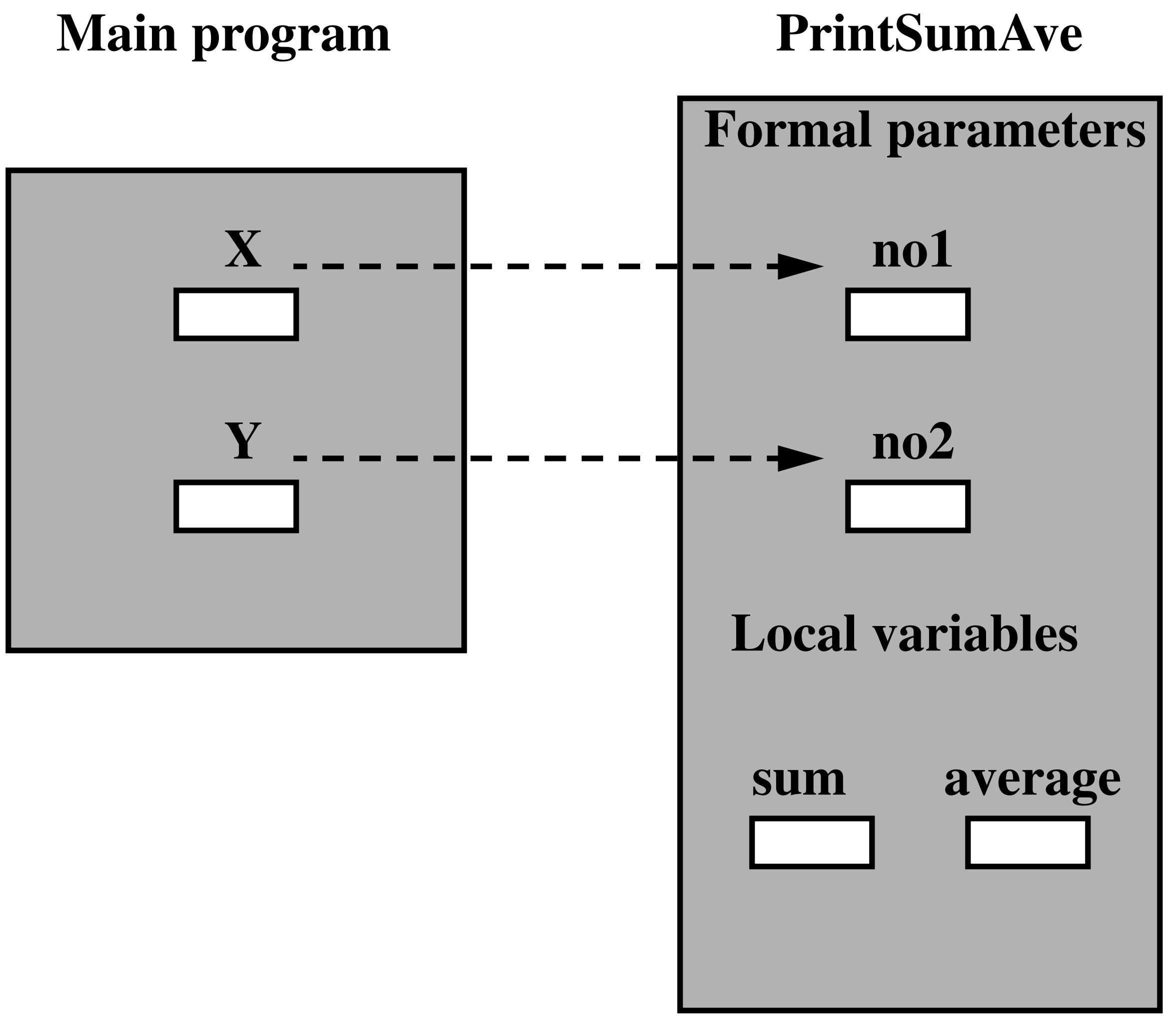
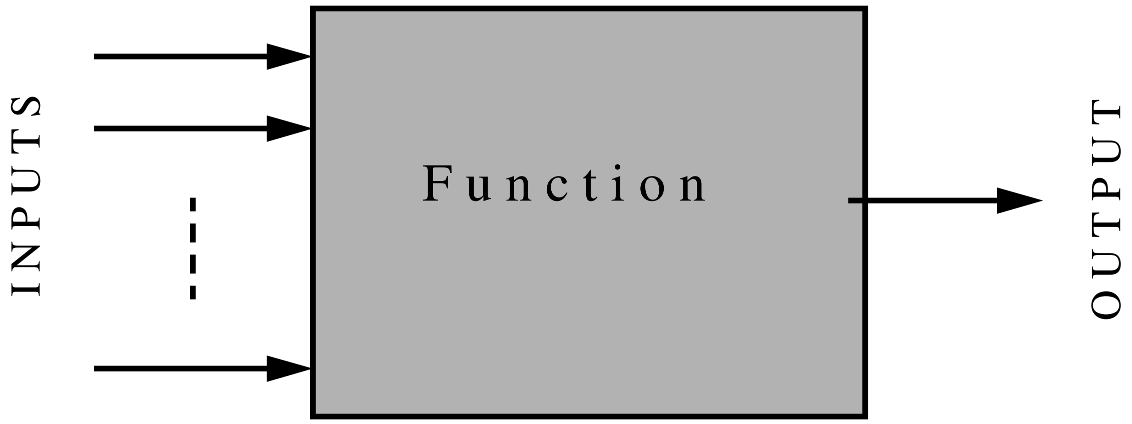
Example:

```
// Print the sum and average of two numbers
// Input: two numbers x & y
// Output: sum - the sum of x & y
//             average - the average of x & y
#include <iostream.h>

void PrintSumAve ( float, float );

void main ( ) {
      float x, y;

      cout << "Enter two numbers: ";
      cin >> x >> y;

      PrintSumAve ( x , y );
}

void PrintSumAve (float no1, float no2) {
      float sum, average;

      sum = no1 + no2;
      average = sum / 2;

      cout << "The sum is " << sum << endl;
      cout << "The average is " << average << endl;
}
```

Data areas after call to PrintSumAve:



Function with inputs and single output:



- <fn_type> <function_id> (<parameter list>) {
        < local declarations >
        < sequences of statements > ;
  }

- Function "function_id" returns a single result of type "fn_type".

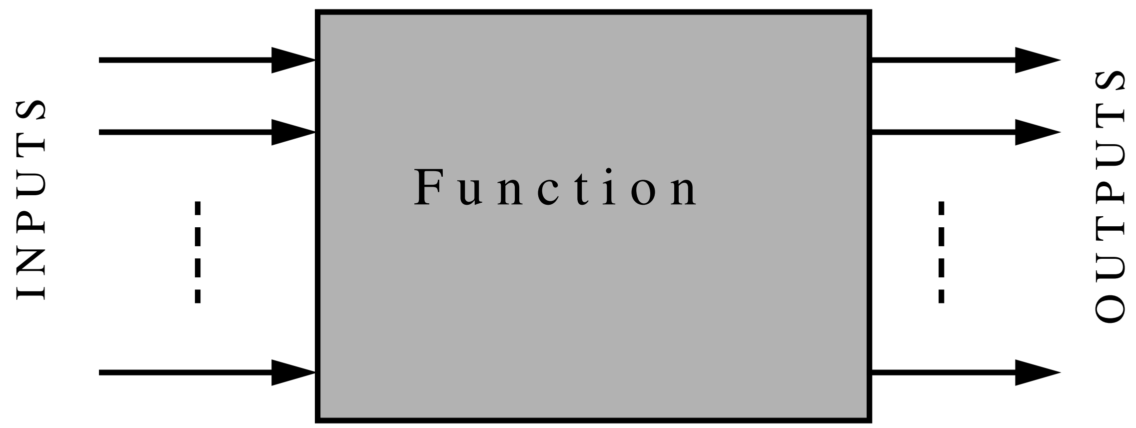
- One of the statements has the form:

    return < expression > ;

Example:

```
// Compute the  average of two numbers
// Input: two numbers x & y
// Output: the average of x & y
#include <iostream.h>

float  average ( float, float );

void main ( ) {
	float x, y;

	cout << "Enter two numbers: ";
	cin >> x >> y;

	ave = average(x, y);
	cout << "The average is " << ave << endl;
}

float average (float no1, float no2) {
	float sum;

	sum = no1 + no2;
	return  sum / 2 ;
}
```

Function with Multiple Outputs



pass-by-reference

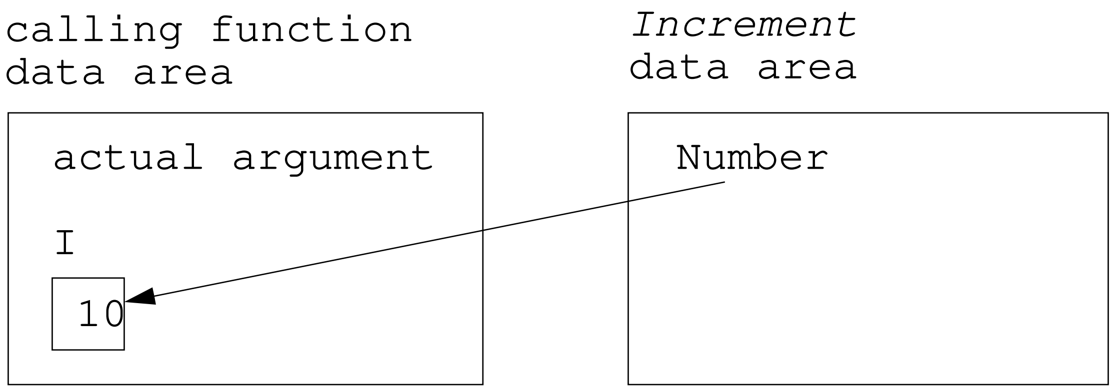
- pass-by-reference
  - reference (address) to actual argument is ***passed*** to the function
  - function manipulates ***actual argument***
- charateristics
  - changes are reflected in actual argument
  - when function terminates, actual argument may be changed
- argument list

  <type>**&** <identifier>, ... , <type>**&** <identifier>

Pass-by-Reference

- Example:

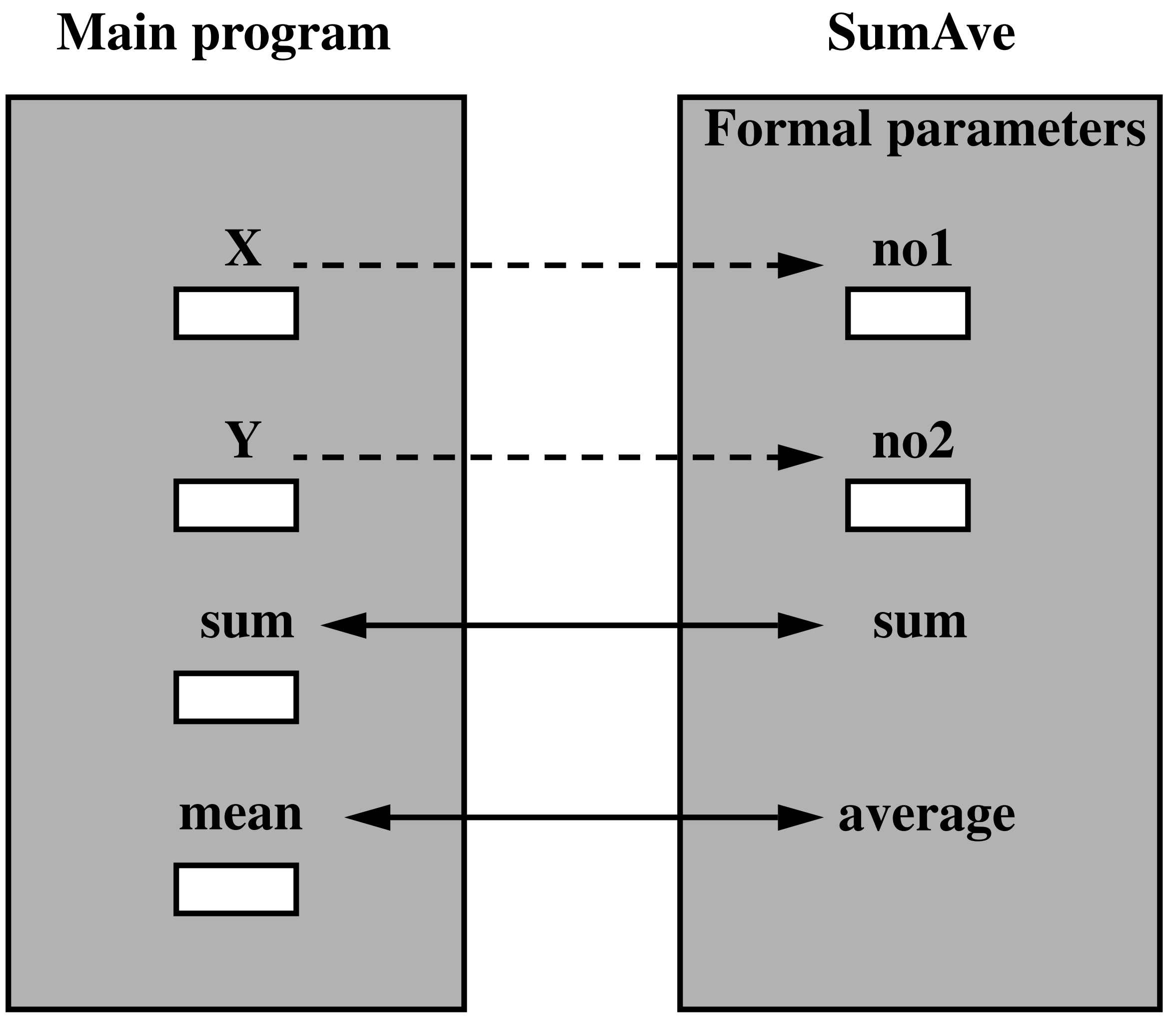
```
#include <iostream.h>

void Increment(int& Number) {
    Number = Number + 1;
    cout << Number << endl;
  }

void main() {
  int I = 10;

  //actual argument is a variable
  Increment(I);
  cout << I << endl;
}
```



Example:

```
// Print the sum and average of two numbers
// Input: two numbers x & y
// Output: sum - the sum of x & y
//            average - the average of x & y
#include <iostream.h>

void SumAve ( float, float, float&, float& );

void main ( ) {
      float x, y, sum, average;

      cout << "Enter two numbers: ";
      cin >> x >> y;

      SumAve ( x , y, sum, average );
      cout << "The sum is " << sum << endl;
      cout << "The average is " << average << endl;
}

void SumAve (float no1, float no2
                        float& sum, float& average)
{
      sum = no1 + no2;
      average = sum / 2;
}
```

Date areas after call to SumAve:



```
#include <iostream.h>

void swap ( int& , int& );

void main ( ) {
	int first, second, third;			// input numbers

	// Read in first, second and third.
	cout << "Enter three numbers: ";
	cin >> first >> second >> third;

	// Compare and swap first and second if needed.
	if (first > second)  swap (first, second);

	// Compare and swap second and third if needed.
	if (second > third) swap (second, third);

	// Compare and swap first and second if needed.
	if (first > second) swap (first, second);

	// Output the sorted numbers first, second and third.
	cout << "The sorted numbers are "
		<< first << " , " << second << " , " << third << endl;
}

//  Function for swapping two numbers
void swap ( int& x,  int& y) {
	int temp;

	temp = x;
	x = y;
	y = temp;
}
```