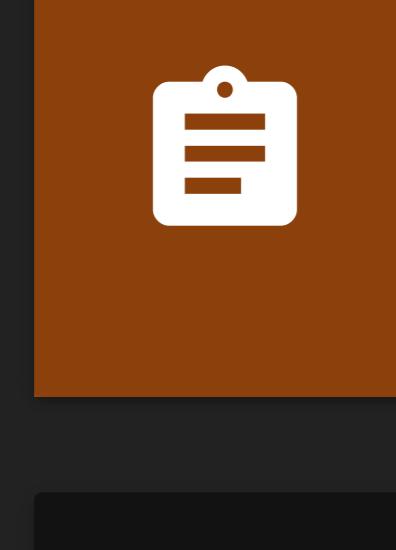


5.10 Pass by reference



Students:
Section 5.11 is a part of 1 assignment: **CSC108 CH05.8-5.16 P5B**

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Includes: PA
Due: 04/15/2025, 11:59 PM EDT

5.11 Scope of variable/function definitions

The name of a defined variable or function item is only visible to part of a program, known as the item's **scope**. A variable declared in a function has scope limited to inside that function. In fact, because a compiler scans a program line-by-line from top-to-bottom, the scope starts *after* the declaration until the function's end. The following highlights the scope of local variable cmVal.

Figure 5.11.1: Local variable scope.

```
#include <iostream>
using namespace std;

const double CM_PER_IN = 2.54;
const int IN_PER_FT = 12;

/* Converts a height in feet/inches to centimeters */
double HeightFtInToCm(int heightFt, int heightIn) {
    int totIn;
    double cmVal;

    totIn = (heightFt * IN_PER_FT) + heightIn; // Total inches
    cmVal = totIn * CM_PER_IN; // Conv inch to cm
    return cmVal;
}

int main() {
    int userFt; // User defined feet
    int userIn; // User defined inches

    // Prompt user for feet/inches
    cout << "Enter feet: ";
    cin >> userFt;

    cout << "Enter inches: ";
    cin >> userIn;

    // Output the conversion result
    cout << "Centimeters: ";
    cout << HeightFtInToCm(userFt, userIn) << endl;
    return 0;
}
```

[Feedback?](#)

Note that variable cmVal is invisible to the function main(). A statement in main() like newLen = cmVal; would yield a compiler error, e.g., the "error: cmVal was not declared in this scope". Likewise, variables userFt and userIn are invisible to the function HeightFtInToCm(). Thus, a programmer is free to define items with names userFt or userIn in function HeightFtInToCm().

A variable declared outside any function is called a **global variable**, in contrast to a *local variable* declared inside a function. A global variable's scope extends after the declaration to the file's end, and reaches into functions. For example, HeightFtInToCm() above accesses global variables CM_PER_IN and IN_PER_FT.

Global variables should be used sparingly. If a function's local variable (including a parameter) has the same name as a global variable, then in that function the name refers to the local item and the global is inaccessible. Such naming can confuse a reader. Furthermore, if a function updates a global variable, the function has effects that go beyond its parameters and return value, known as **side effects**, which make program maintenance hard. Global variables are typically limited to const variables like the number of centimeters per inch above. Beginning programmers sometimes use globals to avoid having to use parameters, which is bad practice. *Good practice is to minimize the use of non-const global variables.*

PARTICIPATION ACTIVITY | 5.11.1: Variable/function scope.

- 1) A local variable is declared inside a function, while a global is declared outside any function.
 - True
 - False
- 2) A local variable's scope extends from a function's opening brace to the function's closing brace.
 - True
 - False
- 3) If a programmer declares a function's local variable to have the same name as a function parameter, the name will refer to the local variable.
 - True
 - False
- 4) If a programmer declares a function's local variable to have the same name as a global variable, the name will refer to the local variable.
 - True
 - False
- 5) A function that changes the value of a global variable is sometimes said to have "side effects".
 - True
 - False

[Feedback?](#)

A function also has scope, which extends from its definition to the end of the file. Commonly, a programmer wishes to have the main() definition appear near the top of a file, with other functions definitions appearing further below, so that the main function is the first thing a reader sees. However, given function scope, main() would not be able to call any of those other functions. A solution involves function declarations. A **function declaration** specifies the function's return type, name, and parameters, ending with a semicolon where the opening brace would have gone. A function declaration is also known as a **function prototype**. The function declaration gives the compiler enough information to recognize valid calls to the function. So by placing function declarations at the top of a file, the main function can then appear next, with actual function definitions appearing later in the file.

Figure 5.11.2: A function declaration allows a function definition to appear later in a file.

```
#include <iostream>
#include <cmath> // To use "pow" function
using namespace std;

/* Program to convert given-year U.S. dollars to current dollars, using simplistic method of 4% annual inflation.
   Source: http://inflationdata.com (See: Historical) */

// (Function DECLARATION)
double ToCurrDollars (double pastDol, int pastYr, int currYr);

int main() {
    double pastDol; // Starting dollar amount
    double currDol; // Ending dollar amount (converted value)
    int pastYr; // Starting year
    int currYr; // Ending year (converted to year)

    // Prompt user for previous year/dollar and current year
    cout << "Enter current year: ";
    cin >> currYr;
    cout << "Enter past year: ";
    cin >> pastYr;
    cout << "Enter past dollars (Ex: 1000): ";
    cin >> pastDol;

    // Function call to convert past to current dollars
    currDol = ToCurrDollars(pastDol, pastYr, currYr);

    cout << "$" << pastDol << " in " << pastYr;
    cout << " is about $" << currDol << " in ";
    cout << currYr << endl;

    return 0;
}

// (Function DEFINITION)
// Function returns equivalent value of pastDol in pastYr to currYr
double ToCurrDollars (double pastDol, int pastYr, int currYr) {
    double currDol; // Equivalent dollar amount given inflation

    currDol = pastDol * pow(1.04, currYr - pastYr);

    return currDol;
}
```

```
Enter current year: 2015
Enter past year: 1970
Enter past dollars (Ex: 1000): 10000
$10000 in 1970 is about $58411.8 in 2015
(Note: Average annual U.S. income in 1970)
...
Enter current year: 2015
Enter past year: 1970
Enter past dollars (Ex: 1000): 23000
$23000 in 1970 is about $134347 in 2015
(Note: Average U.S. house price in 1970)
...
Enter current year: 2015
Enter past year: 1933
Enter past dollars (Ex: 1000): 37
$37 in 1933 is about $922,435 in 2015
(Note: Cost of Golden Gate Bridge, in millions)
...
Enter current year: 2015
Enter past year: 1969
Enter past dollars (Ex: 1000): 25
$25 in 1969 is about $151,871 in 2015
(Note: Cost of Apollo space program, in billions)
```

[Feedback?](#)

A common error is for the function definition to not match the function declaration, such as a parameter defined as double in the declaration but as int in the definition, or with a slightly different identifier. The compiler detects such errors.

PARTICIPATION ACTIVITY | 5.11.2: Function declaration and definition.

- 1) A function declaration lists the contents of a function, while a function definition just specifies the function's interface.
 - True
 - False
- 2) A function declaration enables calls to the function before the function definition.
 - True
 - False

[Feedback?](#)

Exploring further:

- [More on Scope](#) from msdn.microsoft.com

How was this section? [Provide section feedback](#)

0 / 56 points

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Activity summary for assignment: CSC108 CH05.8-5.16 P5B

Due: 04/15/2025, 11:59 PM EDT

0 / 56 points

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Completion details ▾

5.12 Default parameter values