Scope of Variables
The Concept of Scope
Author: Krishna Mahavadi
Concept of Scope

• Scope – The dictionary definition of scope is the extent or range of view.
• Where we declare a variable inherently determines where it would be accessible in a C++ program.
Declaring Variables in a Block

• A block of code is a code that is between { }.
• Variables declared in for and while loops, and if statements in blocks of code are not accessible anywhere else in the program.
• For example, for(int i; i<=n; i++) cout << i;
• Later in the code if you decide to output the last i in a different part of the program out of the for loop, the variable doesn’t exist so the compiler throws an error.
Declaring Variables in a Function

- Variables declared in a function are confined to that function.
- They can’t be shared with other functions.
- Each function needs to be a self sufficient block of code that does a single task.
- All required variables are declared in the function block in order to achieve this self sufficiency.
Passing variables as Arguments

• Any variable that needs to be passed to a function needs to be an argument of the function.
• Only the value of the variable is passed through.
• That’s why we can give it a different name.
• That’s also the reason why the changes we make to the new, different variable doesn’t affect the main program.
#include <iostream>
using namespace std;

void update(int x){
    cout << "Passed in value of x is: " << x << endl;
    x = 0;
    cout << "The new value of x is: " << x << endl;
}

int main() {
    int n = 100;
    cout << "The value of n is: " << n << endl;
    update(n);
    cout << "The value of n is: " << n << endl;
    return 0;
}