Local and global scopes

- constants and variables can be defined inside functions (including main) or outside functions
- local variables/constants: declared inside a function
- global variables/constants: declared outside a function
- locals are only visible inside their function (after declaration)
- globals are visible everywhere in the program (after declaration)
- we'll talk about additional scopes later in the course

Global variables/constants

- declared outside all functions, visible everywhere afterward
- global constants widely regarded as acceptable (e.g. to ensure the use of the same value of Pi throughout a program)
- global variables generally to be avoided (they complicate maintenance and debugging, since any function in the program could be manipulating them)

Globals: example

```
#include <iostream>
using namespace std;
const double Pi = 3.14;
float answer = 0; // final result
float input = 0; // user input

// all the functions access the global vars
// makes trace/debug difficult in long code
void getData();
void calcAnswer();
void showAnswer();
```

We cannot see the use/exchange of data between the functions without manually searching through each code segment

```
int main()
{
   getData();
   calcAnswer();
   showAnswer();
}

void getData()
{
    cout << “Enter your value”;
    cin >> input;
}

void calcAnswer()
{
    answer = Pi * input * input;
}
```

Local variables/constants

- are declared inside a function
- are only visible within that function
- a function doesn't know or care about the locals used by other functions, they're entirely separate
- it's possible for every function to have a local variable they call x, but they're all separate: function A's x, function B's x, etc

Locals: example

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

int getInteger();
float getFloat();

int main()
{
    int a = getInteger();
    float b = getFloat();
    // a,b exist only in main
    cout << a << endl;
    cout << b << endl;
}
```

```
int getInteger()
{
    int userInput; // exists only in getInteger()
    cout << “Enter an integer”;
    cin >> userInput;
    return userInput;
}

float getFloat()
{
    float userInput; // exists only in getFloat();
    cout << “Enter a float”;
    cin >> userInput;
    return userInput;
}
```