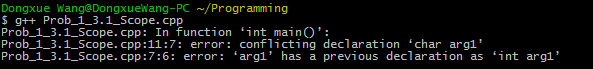
3.1 What happens if we declare the same name twice within a block, giving it two different meanings? Hints: Did your program compile? If so, what does it print? If not, what error message do you get?

Answer: The program doesn’t compile. Please refer to the pic below for error message.



3.2 What happens if we declare an identiﬁer in a block, and then redeclare that same identiﬁer in a block nested within that block?

Answer: It depends on where the identifier name is accessed. If it is accessed within the inner block, the declaration in the inner block is used. If it is accessed within the outer block, the declaration of the outer block is used.

3.3 Suppose an identiﬁer has two diﬀerent declarations, one in an outer block and one in a nested inner block. If the name is accessed within the inner block, which declaration is used?

Answer: The declaration in the inner block is used.

3.4 Suppose an identiﬁer has two diﬀerent declarations, one in an outer block and one in a nested inner block. If the name is accessed within the outer block, but after the inner block, which declaration is used?

Answer: The declaration in the outer block is used.

3.5 Below is a sample program that will not compile. Why not? By moving which line can we get the code to compile?

Answer: It didn’t compile because the line telling the compiler to include “iostream” library appears after the identifier “cout” is called in the “std” namespace in the library. The compiler has to know the library before looking for the identifier “cout” in the “std” namespace. Generally it’s good practice to include the library using the first line of the program.

By moving the last line (#include <iostream>) to the first line, we can get the code to compile.