

Beginning C++ Programming

Compiling Multiple Source Files from the Command-Line

When we compile C++ programs that consist of multiple .cpp source files we must be sure that each is compiled and then all are linked together along with any system libraries to produce an executable program.

If you are working within an Integrated Development Environment (IDE), such as CodeLite, Visual Studio, Xcode, and others, when you add the new classes to your project, the IDE takes care of including them in the build processes.

However, if you are building from the command line, you must make sure that all of the .cpp source files are included on the command-line command that builds your project.

For example, suppose our project consists of:

main.cpp -- the main program driver which includes “Account.h”
Account.cpp -- the Account implementation which also includes “Account.h”
Account.h -- the Account specification

In order to build this project from the command-line you must include all (2 in this case) of the .cpp source files as arguments to the compiler. For example:

```
g++ -Wall -std=c++14 -g main.cpp Account.cpp
```

If you omit any of the .cpp source files then you will surely get linker error.

It is common for students to ‘fix’ the linker errors by using #include “Account.cpp” from main.cpp. This is a bad idea and will lead to all sort of errors later as our projects become larger.

Please do not #include any .cpp source files, you should only #include .h header files.

Please let me know in the Q/A or Private Message if you have any questions or run into any issues building your multiple source file projects from the command-line.

Best regards,
Frank Mitropoulos