

PROGRAMMING MODULE – INTEGRATED DEVELOPMENT ENVIRONMENTS (AKA. IDE) AND COMPILERS



WORKING WITH AN IDE AND C++ COMPILER – SUJAN SARKAR, SRJC CS INSTRUCTOR

Source Code, Object Code, and Executable Code

When a C++ program is written, it must be typed into the computer and saved to a file. A *text editor*, which is similar to a word processing program, is used for this task. The statements written by the programmer are called *source code*, and the file they are saved in is called the *source file*.

After the source code is saved to a file, the process of translating it to machine language can begin. During the first phase of this process, a program called the *preprocessor* reads the source code. The preprocessor searches for special lines that begin with the *#* symbol.

These lines contain commands, or *directives*, that cause the preprocessor to amend or process the source code in some way. During the next phase the *compiler* steps through the preprocessed source code, translating each source code instruction into the appropriate machine language instruction. This process will uncover any *syntax errors* that may be in the program. Syntax errors are illegal uses of key words, operators, punctuation, and other language elements. If the program is free of syntax errors, the compiler stores the translated machine language instructions, which are called *object code*, in an *object file*.

Although an object file contains machine language instructions, it is not a complete program. Here is why. C++ is conveniently equipped with a library of prewritten code for performing common operations or sometimes-difficult tasks. For example, the library contains hardware-specific code for displaying messages on the screen and reading input from the keyboard. It also provides routines for mathematical functions, such as calculating the square root of a number. This collection of code, called the *run-time library*, is extensive. Programs almost always use some part of it. When the compiler generates an object file, however, it does not include machine code for any run-time library routines the programmer might have used. During the last phase of the translation process, another program called the *linker* combines the object file with the necessary library routines. Once the linker has finished with this step, an *executable file* is created. The executable file contains machine language instructions, or *executable code*, and is ready to run on the computer.

Figure 1-5 illustrates the process of translating a C++ source file into an executable file. The entire process of invoking the preprocessor, compiler, and linker can be initiated with a single action. For example, on a Linux system, the following command causes the C++ program named *hello.cpp* to be preprocessed, compiled, and linked. The executable code is stored in a file named *hello*.

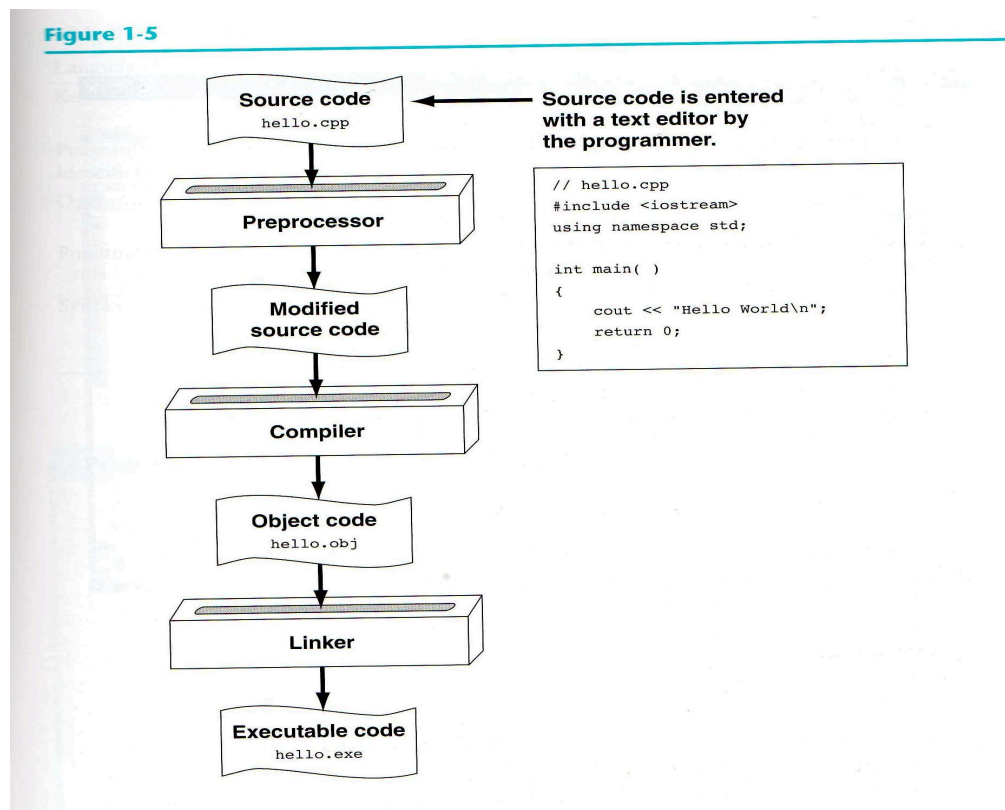
```
g++ -o hello hello.cpp
```

Many development systems, particularly those on person computers, have integrated development environments (IDE's). These environments consist of a text editor, compiler, debugger, and other utilities integrated into a package with a single set of menus. Preprocessing, compiling, linking, and even executing a program is done with a single click of a button, or by selecting a single item from a menu.

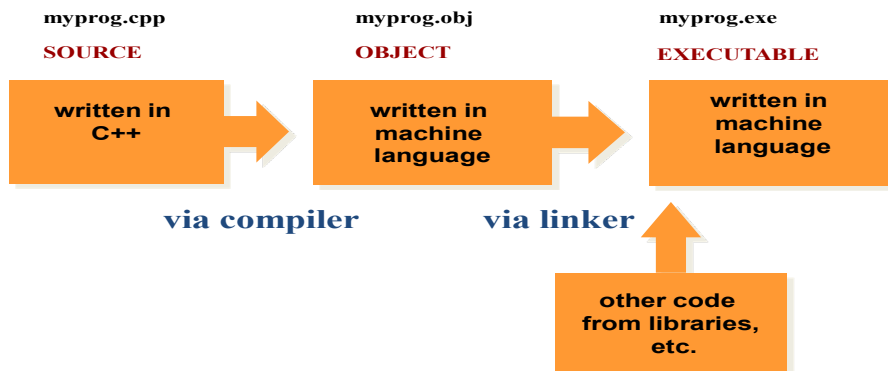
Integrated Development Environments (IDEs) such as

- (1) wxDev-C++: with the completed source code i.e., hello.cpp in the text editor, select the Execute menu item and choose the Compile and Run option (or use the keyboard shortcut F9)
- (2) Code::Blocks: with the completed source code ie., hello.cpp in the text editor, select the Build menu item and choose the Build and Run option (or use the keyboard shortcut F9)
- (3) Microsoft Visual C++: Once you have entered a program's source code, you may compile and execute it by either Pressing Ctrl+F5 or click Debug on the menu bar and select Start without Debugging

The primary content above and the Figure 1-5 below is from Starting out with C++: From Control Structures to Objects by Tony Gaddis see pg 11-12 (adopted text-CS10 summer 2012)



Three C++ Program Stages



WORKING WITH AN IDE AND C++ COMPILER – DEMONSTRATION

1. OPEN EITHER CODE::BLOCKS OR wxDEV-C++
2. CREATE A NEW PROJECT AND SOURCE FILE ...SAVE
3. TYPE IN C++ SOURCE CODE IN THE TEXT EDITOR
4. IN CODE::BLOCK SELECT BUILD>BUILD & RUN OR
5. IN wxDEV-C++ SELECT EXECUTE>COMPILE AND RUN
6. VERIFY IF THE CONTENTS IN THE CONSOLE OUTPUT WINDOW IS CORRECT
7. IF NOT, RETURN TO THE TEXT EDITOR AND MAKE THE NECESSARY CHANGES
8. RECOMPILE THE UPDATED VERSION OF YOUR SOURCE CODE.CORRECT OUTPUT WITH NO ERROR MESSAGES MEANS YOUR PROGRAM IS A SUCCESS! SAVE YOUR WORK!

See CCCConfer Zoom webcast link below

If you are not familiar with CodeBlocks, preview a prior CS10A lecture/lab session introducing a C++ compiler - Zoom Channel Link:

https://santarosa-edu.zoom.us/rec/share/99xSEvbCrXNOB7f9zFHud7UCR6vaeaa81ydL_qALz0hfge9iVQMIWvSG_KeEqH4S?startTime=1578935108000 (*CodeBlocks demo is around the 35-minute mark)

For additional information on “free” Integrated Development Environments (IDE) and compilers see the link below....

<https://canvas.santarosa.edu/courses/63832/pages/download-install-and-set-up-your-ide-slash-cplusplus-compiler>

To practice using a C++ compiler and for a programming challenge click on the link below

https://canvas.santarosa.edu/courses/63832/assignments/959032?module_item_id=2501022

PROGRAMMING CHALLENGE: C++ source code write-up and compile test activity

- (a) Using your IDE/Compiler's text editor write a program to print out the following lines from Dr. Seuss's *Horton Hatches the Egg*.¹

I meant what I said
And I said what I meant
An elephant's faithful
One hundred percent

When you have completed writing the necessary source code, compile and run your program. Do you see the exact quotation on the screen? Do you see a spelling error? If so, go back to your source code and correct the error. Compile and run the program again.

- (b) Put a border of asterisks around the entire quotation (all four sides). Each line of the quotation should be sent to the output stream in the same statement. (This is a bit more of an advanced challengegive it a go)
-

```
/* Program Seuss prints out quotations from Dr. Seuss's Horton Hatches the Egg to  
the computer screen.*/
```

```
//Type your name here
```

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

```
int main()  
{
```

```
    //program statements output the first two lines
```

```
    cout<<"I meant what I said"<<endl;  
    cout<<"And I said what I maent"<<endl;
```

```
    //type in the remaining statements below
```

```
        return 0;  
}
```

¹ Dr. Seuss, *Horton Hatches the Egg* (New York: Random House, 1940).