Assignment 0
Compiler Training and Program Submission
(25 points)

Due: Friday, September 1 on Blackboard by 11:59 PM

Overview

The purpose of this assignment is to learn how to use a C++ compiler and write a short C++ program.

There are a lot of different C++ compilers available to write and run C++ program. The compilers that we recommend for this course are:

- **Windows PC:** Dev C++
- **Mac:** XCode
- **Online:** onlinegdb

If you're familiar with another C++ compiler, it may also be used to complete the assignment. Just make sure to use the sample program and other code from one of the three sections (some of it is found in later steps) and use it with the other compiler.

Once all the steps have been completed, submit the source code file that was created (**the one with the CPP extension**) on Blackboard.

Familiarize yourself with one of the three compilers by completing the steps listed in ONE of the following sections based on the type of computer that you're using:

Windows

The Dev C++ program can be downloaded at:

http://sourceforge.net/projects/orwelldevcpp/

To download, click on the green Download button. The download should automatically start. After the download is finished, double click the file that was just downloaded in order to Run the program that was downloaded (it could take a little bit for the program to start running, so be patient).

Select the language for your install and click the OK button. After agreeing to the license agreement, determine the type of install that you want to do. The easiest install is to select Full. If you don't want the full install, select Custom from the dropdown list and then make sure that:

- TDM-GCC x64 4.9.2 compiler
- Associate C and C++ files to Dev-C++

are checked. If you want some of the other things, you can check those too. Once you have the options selected, click the Next button. Change the destination to where you would like the program to be installed, this could also include a flash drive. Click the Install button.

After the program has installed, click the Finish button.

Running a Program in Dev-C++

The first time that Dev-C++ is run, a few questions will have to be answered. Click Next on the first screen. Select No and then click Next on the second screen. Click on Ok.

1. Start Dev-C++.
2. Select Tools/Compiler Options. Click on the General tab in the window that opens.
3. Make sure that the checkbox next to "Add the following commands when calling the compiler:" is checked.
4. Add the following to the box that is below the checkbox:

   -std=c++14

5. Click the OK button. The compiler should now be ready for your first program.
6. To create a new C++ program, go to File/New/Source File. A blank window will open.
7. Type in the following C++ source code/program in the window, making sure to fill in the Programmer, Section, and Date Due:

```
/****************************************************************
CSCI 240         Program 0     Fall 2023

Programmer:

Section:

Date Due:

Purpose: Compiler training
****************************************************************/

#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
cout << "hello, world";

return 0;
}
```

   Note: If you're working in a computer lab on the NIU campus, you'll have to change the last part of code (the return 0;) to the following:

```
cout << endl;
system("pause");

return 0;
```

   This will keep the output window from closing until Enter is pressed.

8. Save the file. *Use File/Save*. You will see the pop-up *Save File* dialog window. You should select the location where you want your program to be saved. It could be your hard drive on your own PC (DO NOT save the file directly on the C: drive, it must be within a folder/directory on the C: drive if that's where you decide to save the file), your network drive (in the labs) or any subdirectory in any of these locations. Be sure you know where you are saving the file so you can find it tomorrow.

   Choose a meaningful file name with a .cpp extension. For example, the program above might be named *hello.cpp*

   Notice that once you have saved the file with a .cpp extension, some of the words in your program are now in color. For example, the bold words are C++ *reserved* words that have special meanings. Later, you can set special colors for other kinds of items in your program (e.g. strings, comments) using *Tools/Options/Edit*. Doing this is optional.

9. Now *compile* the program. Choose *Execute/Compile*. You will see a pop-up window and after a few seconds, the Status should read "Done in ___ seconds". If it does not, you have a compile error on the line that is highlighted. You must find and fix all such errors, and re-compile until you get the "Done in ___ seconds" message.

   Once you have the "Done in ___ seconds" message, *Close* the pop-up window.

10. Now run the program by choosing *Execute/Run*. Depending on the nature of your program, you may see a prompt for user input (which you provide by typing on the keyboard) or some output displayed on the black screen (DOS window). If you typed the program listed above, you will see "hello, world"

    When the last instruction in your program has executed, Dev-C++ will display the message "Process exited after ___ seconds with return value 0. Press any key to continue...". When you do this, the DOS window will disappear.

    Note: If this run step causes the error message "Couldn't create process" to show up. Choose Tools/Compiler Options. On the Compile tab, change the Compiler set to configure option to the opposite value (ie. if it's currently, 64-bit, change it to 32-bit). Re-compile the program and try to run it again.

11. Congratulations. You have created and run your first C++ program.

    Sit back and relax for 15 seconds... now let's go on.

    First a couple of notes on what you have just done.

    When you did *Execute/Compile* and *Run*, notice that on the Execute menu there are shortcuts shown: F9 for *Compile* and F10 for *Run*. You can use these keys as shortcuts for the corresponding mouse selections. (F11 will do both steps)

    When you are working in Dev-C++, you will need to keep track of several windows: we've seen the pop-up window for Compile, and the DOS window, as well as the window for Dev itself and the editing window within Dev where you type and edit your program. All of these work together and you will need to mentally keep track of which ones are open and where they are (recall, for example, that sometimes the DOS window is minimized as a button on the Task Tray). Sometimes one window may be hidden behind another. We assume that you have used Windows enough that you are familiar with the basics of manipulating the various windows. If not, see one of the 240 TA's during office hours for a brief orientation.

12. Now, just to see what will happen, let's make a couple of mistakes on purpose. Use the delete key to erase the closing " (double quote character) after the word *world* in your program. Now try to compile it (*F9*). Notice that the dialog reports several errors. This is common - one mistake may cause several error messages. Your usual response will be to try to fix the first error in your program (which is usually indicated by the first or first few messages) and re-compile. In this case, the first error mentions "missing terminating " character" and a line number (line 20 if you typed in the program exactly as shown) which will usually indicate the location of the error.
13. Now try deleting the semicolon at the end of the same line. Compile. You see the same error messages even though we know there are now **two** errors. This often happens: one error "hides" another. Restore the " but leave the ; missing. Recompile. Now you see the message "error: expected ';' before 'return'".
14. Restore the ; and change the character string to: "\nhello, world" That's a backslash before the n. Compile and run the program. Notice there is now a blank line at the top of the DOS screen. The sequence "\n" makes the cursor on the DOS screen go down one line. (Try it again using a forward slash to see what will happen.)
15. Misspell the word main. Make it "maain". Compile. A Linker error should show up at the bottom of the screen. Read it. This illustrates that some error messages are not entirely clear.
16. Fix the program, compile and run it again.
17. (Almost) finally, before you leave, save the program. You will normally want to make a backup (a separate) copy of your work in case the the original is lost due to equipment failure (it could happen) or some mistake on your part (also could happen).
18. Choose *File/Exit* to quit your Dev session and close Dev. You can then do other work on the computer. First, though, pretend you have come back (another day maybe) to continue working on your program.
19. Start Dev again. Use *File/Open* and locate the file you want to work on (i.e. in this case, the one you just saved in step 17). Add another line after the first *cout* line but before the *return* line:

```
cout << "\ngoodbye for now..";
```

    Compile, run, check it, save, and quit.

    That's enough for one day.

20. Submit the .cpp file on Blackboard for grading. As noted earlier, make sure that the Programmer, Section, and Date Due have been filled in in the box at the top of the program.

Mac

The XCode program can be downloaded at:

https://developer.apple.com/xcode/

Download the version that is recommended for the operating system that you're running. **Note:** You're probably going to have to create a free ADC membership in order to download XCode.

1. Start XCode.
2. Choose "Create a new XCode project".
   - On the next screen you have to choose a template for your project. Under OS X, select Application. (For the newer versions of XCode, select Mac OS X.) Select the Command Line Tool icon and click the Next button.
   - Now choose the options for your new project. Enter a value for Product Name - this should be the name that you want for your assignment, for this practice project, just enter Hello. Enter a Company Identifier if it is required - this can be anythig that you want. Make sure that Type is C++ and then click the Next button.
   - Determine where you want the files to be saved and then click the Create button.
   - You should now have a project that contains 2 files: main.cpp and _____.1 where the _____ is the Product Name from earlier. (For the newer version of XCode, it's possible that there will not be a .1 file. That's okay.)
3. Click on the main.cpp file name to open it. Once it opens, it will probably have some C++ code that was generated by XCode. You can either delete this or modify it with your own C++ source code:
4. Type in the following C++ source code/program in the window, making sure to fill in the Programmer, Section, and Date Due:

```
/****************************************************************
CSCI 240         Program 0     Fall 2023

Programmer:

Section:

Date Due:

Purpose: Compiler training
****************************************************************/

#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
cout << "hello, world";

return 0;
}
```

5. Save the file. Use *File/Save*.

   Notice that some of the words in your program are in color. For example, the purple and pink words are C++ *reserved* words that have special meanings.

6. Now *compile and run* the program by clicking on the Run arrow in the upper left corner. If the code does not contain any compile/syntax errors, you will see a "Build Succeeded" message. If the "Build Failed" message shows up, you have a compile error on the line that is highlighted. You must find and fix all such errors, and re-run until you get the "Build Succeeded" message.

   Once you have the "Build Succeeded" message, the program will automatically run and any output that is produced by the program will show up in the "All Output" box below the source code.

   Depending on the nature of your program, you may see a prompt for user input (which you provide by clicking in the All Output box and entering a value by typing on the keyboard) or some output displayed in the window. If you typed the program listed above, you will see "hello, world"

7. Congratulations. You have created and run your first C++ program.

   Sit back and relax for 15 seconds... now let's go on.

8. Now, just to see what will happen, let's make a couple of mistakes on purpose. Use the delete key to erase the closing " (double quote character) after the word *world* in your program. Now try to run it. Notice that the code has been highlighted with 2 errors. This is common - one mistake may cause several error messages. Your usual response will be to try to fix the first error in your program (which is usually indicated by the first or first few messages) and re-run. In this case, the error mentions "missing terminating " character".
9. Now try deleting the semicolon at the end of the same line. Run the program. You should see the same error messages even though we know there are now **two** errors. This often happens: one error "hides" another. Restore the " but leave the ; missing. Re-run. Now you see the message "Expected ';' after expression'".
10. Restore the ; and change the character string to: "\nhello, world" That's a backslash before the n. Run the program. Notice there is now a blank line at the top of the DOS screen. The sequence "\n" makes the cursor on the output screen go down one line. (Try it again using a forward slash to see what will happen.)
11. Misspell the word main. Make it "maain". Run. The "Build Failed" message should show up but no code was highlighted. This is because the error was not a compile/syntax error, it was a linker error. Click on the Issue Navigator (the little triangle with the exclamation point) to see the linker errors. Read them. This illustrates that some error messages are not entirely clear.
12. Fix the program, compile and run it again.
13. (Almost) finally, before you leave, save the program. You will normally want to make a backup (a separate) copy of your work in case the the original is lost due to equipment failure (it could happen) or some mistake on your part (also could happen).
14. Choose *File/Close Project* to quit your XCode session and close XCode. You can then do other work on the computer. First, though, pretend you have come back (another day maybe) to continue on working your program.
15. Start XCode again. Your recent projects should appear in the Recents box. Select the project and click open. In the main.cpp file, add another line after the first *cout* line but before the *return* line:

```
cout << "\ngoodbye for now..";
```

    Run, check it, save, and quit.

    That's enough for one day....Except

    One final note, when the project was created, it was placed in a folder wherever you specified in Step 2. The folder will be named whatever you supplied for Product Name. For the above example, the folder should be named Hello.

    Inside of the Hello folder will be two items: another Hello folder and the file Hello.xcodeproject

    Inside of the second Hello folder, there should be 2 files: main.cpp and Hello.1. The main.cpp file is the one that you will submit when turning in assignments.

16. Submit the .cpp file on Blackboard for grading. As noted earlier, make sure that the Programmer, Section, and Date Due have been filled in in the box at the top of the program.

Online

There are many online C++ compilers for those that can't install a compiler on the device that is being used for the course. Many students from past CSCI 240 courses have recommended the C++ compiler that is available at onlinegdb.com.

https://www.onlinegdb.com/

Set the programming language to C++14 in the dropdown menu that is located above the upper right corner of the editor screen.

Running a Program

1. Type in the following C++ source code/program in the editor window, making sure to fill in the Programmer, Section, and Due Date. You can also modify the code that is given when the onlinegdb.com page loads to match what is below:

```
/****************************************************************
CSCI 240         Program 0     Fall 2023

Programmer:

Section:

Date Due:

Purpose: Compiler training
****************************************************************/

#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
cout << "hello, world";

return 0;
}
```

2. To save a copy of your source code, click the button with the downward pointing arrow that is located above the editor window. This will create a file named ***main.cpp*** in the default location for your browser downloads. An alternative is to click the Save button, but this requires a Google+, Facebook, or Github account to save the program.
3. Make sure the radio button with Interactive Console is selected. This is located below the editor window with the source code. This selection will allow the program to be run with interactive user input (if that is something that the program uses).
4. Now *compile and run* the program. Click the *Run* button that is above the editor window with the source code. After a few seconds, you should see some results. A box with the output that is produced by the source code should appear below the editor window. If it does not, you have a compile error on the line that is specified in the error message. You must find and fix all such errors, and re-run until you get the program output.
5. The output that is produced by the program listed above is "hello, world". When you're done viewing the output, you can hit Enter.
6. Congratulations. You have created and run your first C++ program.

   Sit back and relax for 15 seconds... now let's go on.

   First a recommendation...It is probably a good idea to make it a habit of occasionally downloading your source code as you're developing your programs to ensure that you don't accidentally lose any of your work.

7. Now, just to see what will happen, let's make a couple of mistakes on purpose. Use the delete key to erase the closing " (double quote character) after the word *world* in your program. Now try to compile and run the program. Notice that the Compilation failed due to the following error(s) message displays and there are several errors listed. This is common - one mistake may cause several error messages. Your usual response should be to try to fix the first error in your program (which is usually indicated by the first or first few messages) and re-run. In this case, the first error mentions "missing terminating " character" and a line number (line 20 if you typed in the program as shown) which will usually indicate the location of the error.
8. Now try deleting the semicolon at the end of the same line. Run. You see the same error messages even though we know there are now **two** errors. This often happens: one error "hides" another. Restore the " but leave the ; missing. Re-run. Now you see the message "error: expected ';' before 'return'".
9. Restore the ; and change the character string to: "\nhello, world" That's a backslash before the n. Run the program. Notice there is now a blank line at the top of the output. The sequence "\n" makes the output go down one line. (Try it again using a forward slash to see what will happen.)
10. Misspell the word main. Make it "maain". Run. A Linker error should show up in the Compile Info. Read it. This illustrates that some error messages are not entirely clear.
11. Fix the program, run it again.
12. Download a copy of the source code. Refresh/reload the webpage so that it returns the original default code from when you first loaded the page. To continue working on an existing program, click the Upload File button (it's the one with the cloud and arrow that points upward). Locate the file that was downloaded (probably named main.cpp).
13. Once the file is opened, add another line after the first *cout* line but before the *return* line:

```
cout << "\ngoodbye for now..";
```

    Run, check it, save, and quit.

    That's enough for one day.

14. Submit the .cpp file on Blackboard for grading. As noted earlier, make sure that the Programmer, Section, and Date Due have been filled in in the box at the top of the program.

Running a Program with User Input

Most programs will use some kind of input that will be used in calculations or displayed. To add input to a program that is being run in onlineGDB's compiler, simply type the information in the output window when a cursor appears.