ME 101 Assignment 1 Winter 2022

Deliverables

This assignment will be completed individually

In this assignment you will:

- Compile, debug, and modify a simple C++ program
- Write simple C++ programs

There are four deliverables for this assignment:

- Question 1
- Question 2
- Question 3
- Question 4 – Academic Integrity Quiz (available on UW Learn)

Dev-C++

This course presumes the use of the Dev-C++ integrated development environment. It is not mandatory that you use Dev-C++, but the support staff are most familiar with that environment and may be less able to help resolve issues with others.

Naming Conventions

While Dev-C++ will allow you to name your files and folders anything you like, there can be issues with spaces and special characters.

File and folder names should be eight characters or less, and contain only letters, numbers, and underscore characters.

Remotely using Dev-C++ on the school computers (englab)

Dev-C++ should be installed in all of the engineering computer labs and can be run remoted by logging into one of these computers. Detailed instructions on how to do this are posted as a video on Learn. You will go to https://uwaterloo.ca/engineering-computing/ and follow the instructions for installing a VPN on your computer. Once you are running the VPN, you can login to the engineering computer lab machines by going to https://englab.uwaterloo.ca/.

Due to network security, executables cannot be run from your Documents folder. The recommended process is to initially save your work in a folder under Documents, and then copy the executable (.exe) file to the desktop to run the program.

Using Dev-C++ on the school computers (in person)

Dev-C++ should be installed in all of the engineering computer labs that you can access in person. Due to network security, executables cannot be run from your Documents folder. A list of engineering computer lab locations is available at: https://uwaterloo.ca/engineering-computing/computer-labs

The recommended process is to initially save your work in a folder under Documents, and then copy the executable (.exe) file to the desktop to run the program.

Using Dev-C++ (or another compiler) on your own computer:

If you are working on your own computer, you will need to download Dev-C++. Links and an instruction video are posted on Learn.

Please note that the teaching team has experience supporting the Dev-C++, but it will only work on Windows. If you are using a Mac, or you'd prefer to use a different program, we will try our best to help but it is your responsibility to understand the software you choose. Mac options include Xcode (easier to install) and CodeBlocks (old, need to install gnu tools to run, easier to use once installed).

Question 1

Part 1: Getting the Code to Run

1. Start the compiler by opening a search window and typing Dev-C++, or go to Embarcadero Dev-C++ in the menu and select Dev-C++. Click on the application.
2. Download square.cpp from learn
3. Click on Open Document or File -> Open and browse to find the sample program file square.cpp. Click Open
4. Change the first line so that it includes your name
5. Press F9 or click the ⊞ icon as a shortcut. This “compiles” your code, meaning that it attempts to turn the code you see on the screen into a program that the computer can run. However, there are errors in the code, placed there on purpose so you get experience debugging code
6. You will see several error messages listed in the message window
7. Read the section below about dealing with error messages. Fix all of the errors in the code
8. When all of the errors are gone, the compiler will produce a file with the extension .exe, t his is the executable file which can be run

    - If you are using a school computer, find the .exe file in the folder your code is in, then drag it on to the desktop. Double click it and a new window will open up. It will show you the output of the code you just made

    - If you are using your own computer, you may just click Compile & Run or ▦ and the executable should run directly

Debugging

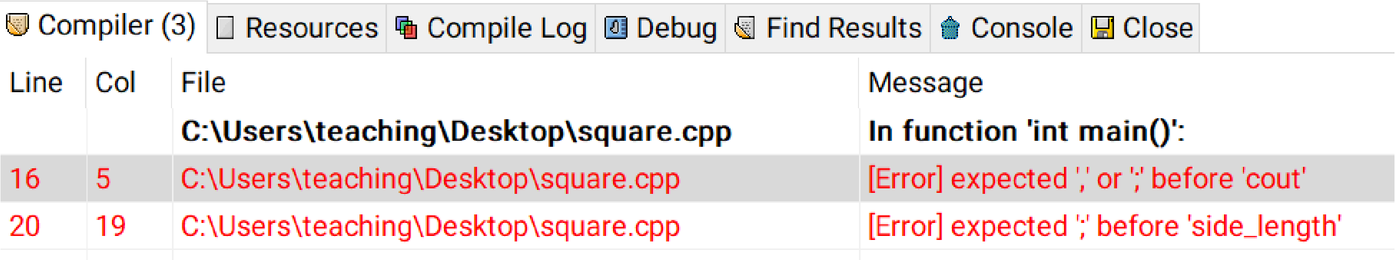
Making errors that prevent compilation is very common, especially at first. Dev-C++ and many other programming environments will tell you two important things about your errors:

1. Where the errors in the code are
2. What the errors might be

Note:

- The compiler can only identify syntax errors
- The compiler usually is very good at telling you *where* the error is
- The compiler usually is not very good at telling you *what* the error is

In the image below, we can see a screenshot of the Dev-C++’s error messages for square.cpp:



Line	Col	File	Message
		C:\Users\teaching\Desktop\square.cpp	**In function 'int main()':**
16	5	C:\Users\teaching\Desktop\square.cpp	[Error] expected ',' or ';' before 'cout'
20	19	C:\Users\teaching\Desktop\square.cpp	[Error] expected ';' before 'side_length'

1. Double-click on the first message. Line 16 in your code should be highlighted.
2. The error is indicating that it expects a `;` before the `cout`.
3. We know that semicolons go at the end of statements, but the error seems to imply it should go before `cout`. This is an example of the compiler not quite getting it right.
4. What it means: the statement before `cout` is missing a semicolon. This is actually a previous line of code
5. Add the missing semicolon at the appropriate location
6. Fix the remaining errors and run your program
7. Note that there may be logical errors. Test your code to ensure the expected output.

Part 2: Getting Input from the User

1. Modify the program to prompt the user to input the value `side_length` before it is squared and cubed. Compile and run the program again.

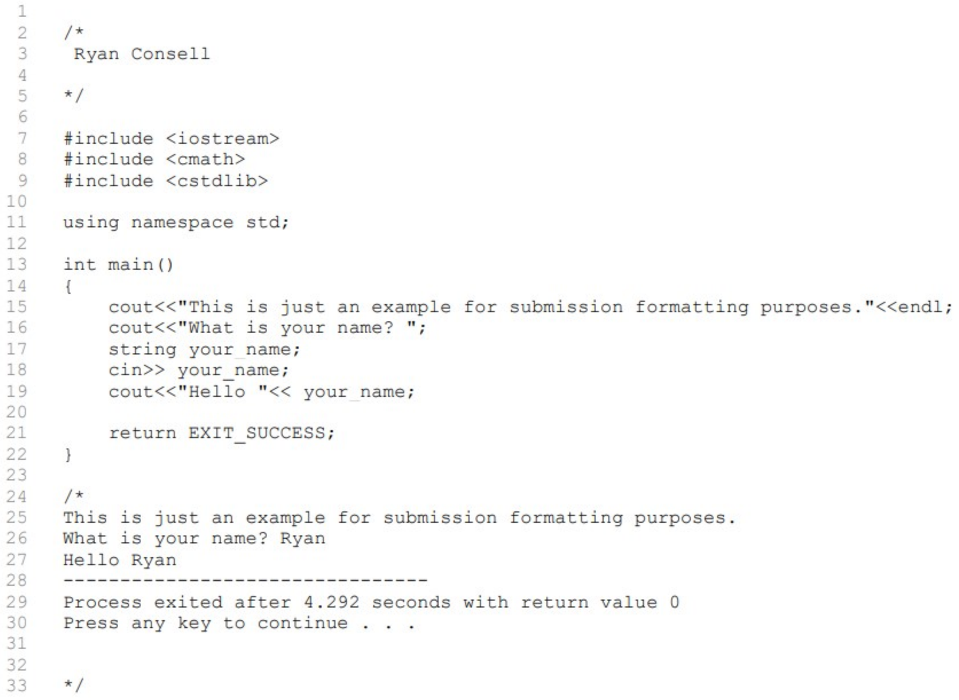
2. To capture any console output, highlight all of the text then copy and paste it.
    a. Depending on your version of windows and settings, you may need to click the top left corner of the console window (black screen) and then click Edit/Mark and use the mouse to select the output. Press the Enter key and your output goes to the clipboard. You can now paste it at the bottom of your program using Ctrl-V

You can use this source file as a template for the remaining questions in this assignment. Make a new source file each time, and copy and paste the contents of this one in to get started

What You Need to Submit into Crowdmark

For every assignment question, you will always be submitting your code and any output that is generated by the code. For question 1 of this week's assignment, you should submit

1. Your modified code from part 2 above
2. The output of your modified code with a test case of side_length = 5, copied and pasted into a `/*block comment*/` at the bottom of your code. An example of the expected format is below.



```

/*
 Ryan Consell

*/

#include <iostream>
#include <cmath>
#include <cstdlib>

using namespace std;

int main()
{
    cout<<"This is just an example for submission formatting purposes."<<endl;
    cout<<"What is your name? ";
    string your_name;
    cin>> your_name;
    cout<<"Hello "<< your_name;

    return EXIT_SUCCESS;
}

/*
This is just an example for submission formatting purposes.
What is your name? Ryan
Hello Ryan
--------------------------------
Process exited after 4.292 seconds with return value 0
Press any key to continue . . .


*/
```

Question 2

Problem Description

The conversion for a temperature given in Fahrenheit to Celsius is given by:

$$Temperature\ ℃ = \frac{5}{9} \times (Temperature\ ℉ - 32)$$

Write a program that:

- Prompts the user for a temperature in Fahrenheit
- Outputs to the screen the temperature in Celsius

What You Need to Submit into Crowdmark

- Run the program for temperatures of 212, 100, 32, -40
- Submit your code with the output from the given test cases pasted at the bottom as a `/*block comment*/`

Question 3

Problem Description

A chain is a unit of length, and is equal to 66 feet, or 22 yards. One chain can be divided into 100 links or 4 rods. There are 10 chains in a furlong.

Write a program that:

- Prompts the user for a length in chains.
- Outputs to the screen the length in feet, yards, links, rods, and furlongs.

What You Need to Submit into Crowdmark

- Run the program for a length of 25 chains
- Submit your code with the output from the given test case pasted at the bottom as a `/*block comment*/`

Question 4: Academic Integrity Quiz

Login to Learn and complete the Academic Integrity Quiz. You can attempt the quiz an unlimited number of times and the highest grade is used. Most questions have hints to help you if you are stuck on a question.

Submitting Questions 1, 2, and 3 of the Assignment

1. Ensure that your code is properly formatted such that it is easy to read
2. Ensure that your name and any acknowledgments of help are included at the top of the relevant files as `//line comments` or `/* block comments */`
3. Ensure that the output included with your file is generated by the code that is being submitted. You are including the output as proof of how the code runs.
4. Submit all three files to Crowdmark. Crowdmark will only accept pdf, jpg, or png documents. You may be able to print directly to pdf from Dev-C++, but if you are unable to do so, then:
    1. Save all of your code (`.cpp`) files as `.txt` files. Using Dev-C++, you can do this with file->save as, then putting `.txt` at the end of your file name
    2. Open the file in notepad
    3. Print to pdf
    4. Upload to crowdmark and review the uploaded pdf to ensure it looks correct
5. Make sure you save an electronic copy of the `.cpp` files. You are allowed to re-use the code in future ME 101 assignments this term provided it is acknowledged