# Compiling and Running a C++ Program

We include a **Makefile** with each project in this course, which is a kind of script that tells the compiler how to build your program. It gives instructions to the compiler and linker about which source files to use. All you need to do is type **make** on the command line in the appropriate directory. Let's try it out.

If you downloaded the example code as described in the previous reading, you should be able to navigate to the cpp-class example and try compiling and running it like this. (Here, we'll assume you installed everything in the **~/environment/coursera-cs400** directory on your AWS Cloud9 environment.) Here's a picture of what that the terminal looks like. We'll talk about it more below.
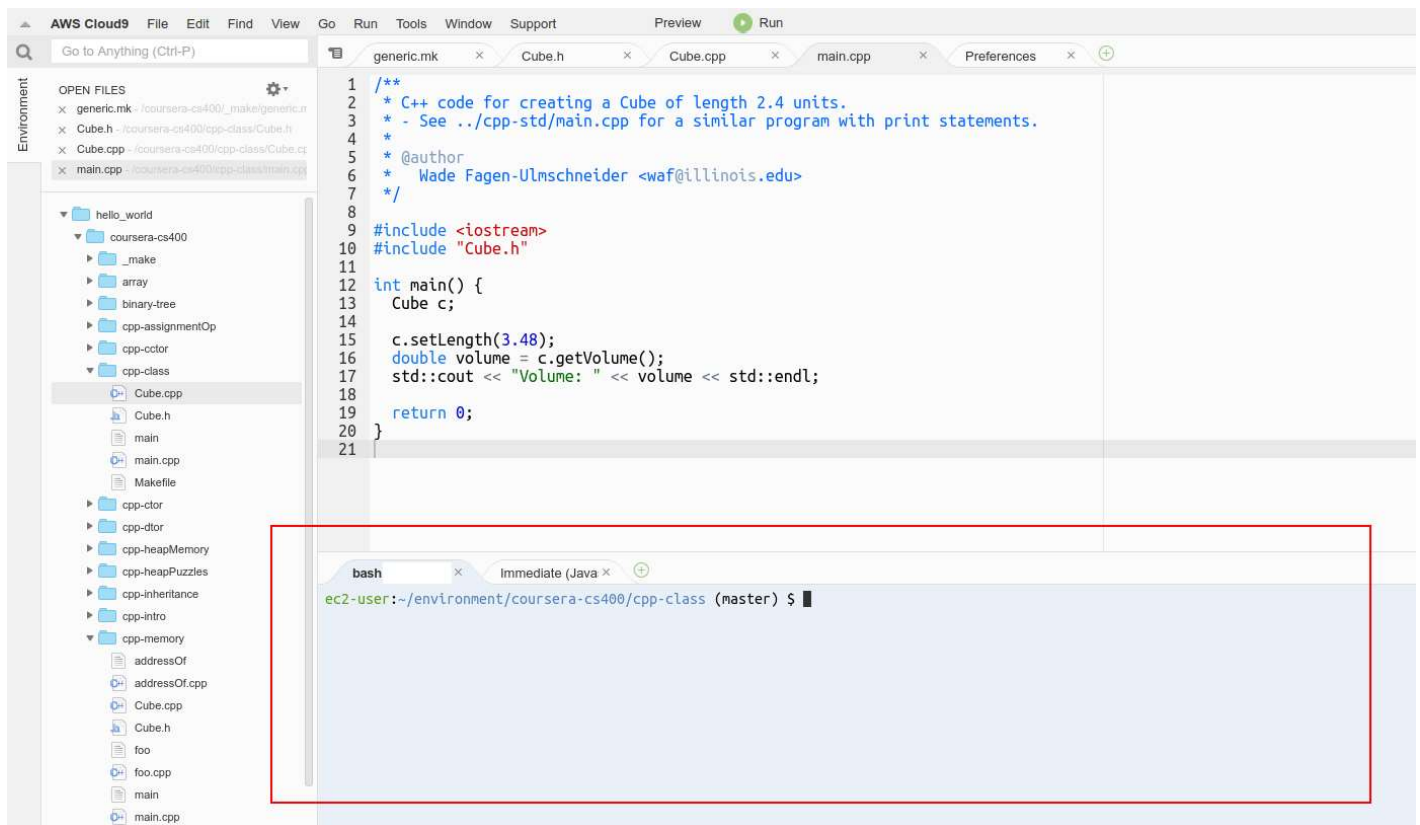
To begin with, the Linux Bash terminal might show this:

```
1    ec2-user:~/environment $
```

This shows that we're already in the ~/environment directory. First, we need to navigate to the directory where our example was downloaded, using the **cd** command to change directory to ~/environment/coursera-cs400/cpp-class as shown:

```
1    ec2-user:~/environment $ cd ~/environment/coursera-cs400/cpp-class
2    ec2-user:~/environment/coursera-cs400/cpp-class (master) $
```

Now, let's list what's already in this directory by typing:

ls

```
1    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ ls
2    Cube.cpp  Cube.h  main.cpp  Makefile
```

We can build the project by typing:

make

```
1    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ make
2    g++ -std=c++14 -O0 -pedantic -Wall -Werror -Wfatal-errors -Wextra -Wno-unused-pai
3    g++ -std=c++14 -O0 -pedantic -Wall -Werror -Wfatal-errors -Wextra -Wno-unused-pai
4    g++ .objs/main.o .objs/Cube.o -std=c++14 -o main
5
6    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ ls
7    Cube.cpp  Cube.h  main  main.cpp  Makefile
```

When we ran make, it triggered several compilation and linking steps and produced an output file called **main**, which is the actual executable program! We can now try running it by typing:

./main

```
1    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ ./main
2    Volume: 42.1442
3    ec2-user:~/environment/coursera-cs400/cpp-class (master) $
```

That ran our main program, produced some text output in the terminal, and returned to the command prompt. The **./** part of **./main** just means "in this directory". When we want to run a program that is in the current directory, we have to specify that by writing **./** in front of the name. Otherwise, Linux would look for a system-wide command with that name instead. Notice that this doesn't work if we try it without the **./** part:

```
1    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ ls
2    Cube.cpp  Cube.h  main  main.cpp  Makefile
3    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ main
4    bash: main: command not found
5    ec2-user:~/environment/coursera-cs400/cpp-class (master) $
```

If you want to clear out all the compiled object and executable files in order to ensure that your program gets recompiled from scratch next time, you can type:

make clean

```
1    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ ls
2    Cube.cpp  Cube.h  main  main.cpp  Makefile
3    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ make clean
4    rm -rf main  .objs  *.o *.d
5    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ ls
6    Cube.cpp  Cube.h  main.cpp  Makefile
```

In this case, the "make clean" script deleted the "main" executable file and the hidden .objs subdirectory.

Before we finish this lesson, here's another terminal tip. You can go "up" one directory from your current location by typing:

cd ..

(You must include the two dots!)

```
1    ec2-user:~/environment/coursera-cs400/cpp-class (master) $ cd ..
2    ec2-user:~/environment/coursera-cs400 (master) $
3
```

And you can also go down into directories by specifying them relative to the current directory. For example, when we're in the ~/environment directory, we can see that there's a coursera-cs400 directory right beneath us, so we can use **cd coursera-cs400** to go directly into it, instead of typing **cd ~/environment/coursera-cs400**:

```
1    ec2-user:~/environment $ ls
2    coursera-cs400  README.md  re_bak
3    ec2-user:~/environment $ cd coursera-cs400
4    ec2-user:~/environment/coursera-cs400 (master) $
```

We hope this helps you work in the terminal!