Southern New Hampshire University

CS 410 Codio Guide

Southern New Hampshire University

Table of Contents

Section 1: Overview of Software Reverse Engineering Using Codio

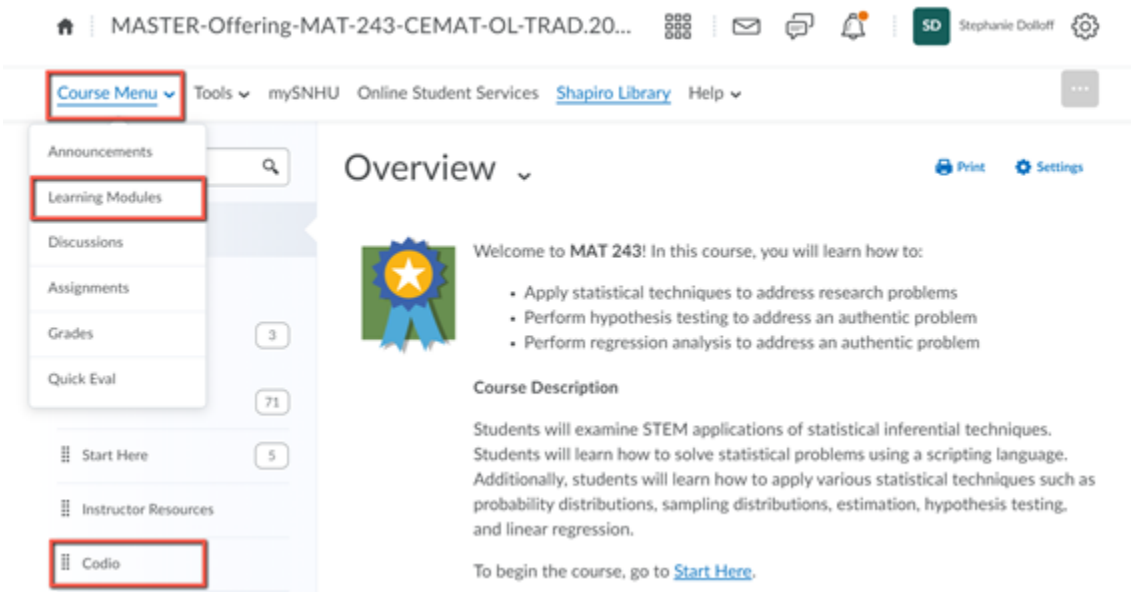
In software reverse engineering, you may only have access to a hex, binary, assembly, or C file, or to a type of source code file. To access these files, there are a variety of tasks you can complete within Codio or use Codio to access an application:

1. Use the Bless hex editor within Codio to open and review hexadecimal files.
2. Use commands within the Codio terminal to compile lower-level programming languages.
3. Write C++ code using Eclipse.
4. Convert the updated code file back to an executable version of the file.

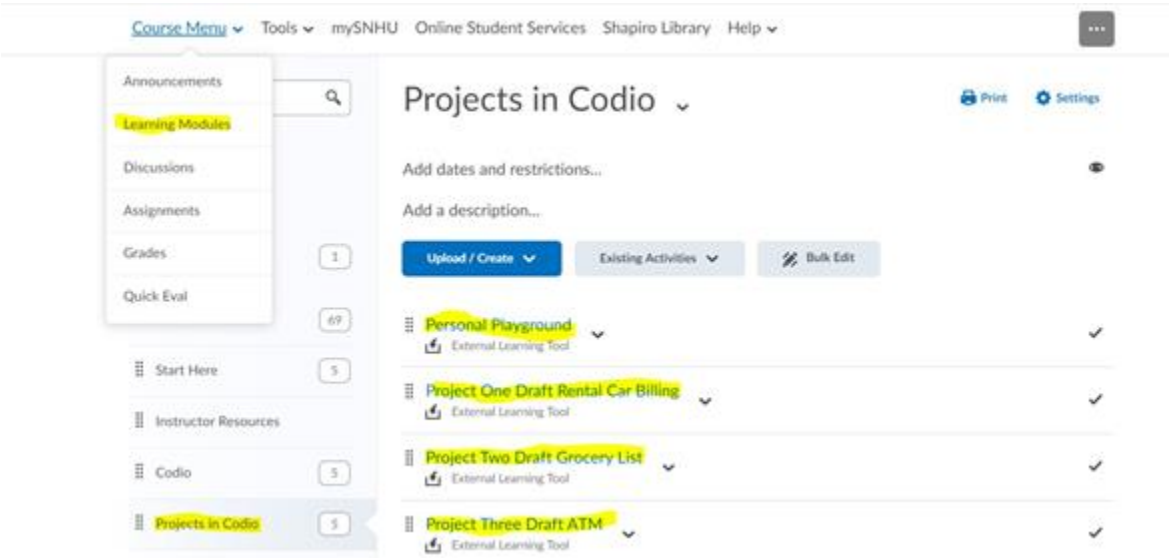
Section 2: Accessing the Virtual Desktop and Eclipse (IDE)

Your Codio environment has an integrated Virtual Desktop and the Eclipse integrated development environment (IDE). These are necessary tools to successfully complete various activities in this course. You should use the Google Chrome browser. The Codio Virtual Desktop is persistent; therefore, your work can be saved to the desktop. The Virtual Desktop should be accessed from Brightspace every time.

1. Access the Virtual Desktop module via Course Menu, then Learning Modules in your Brightspace course navigation menu.

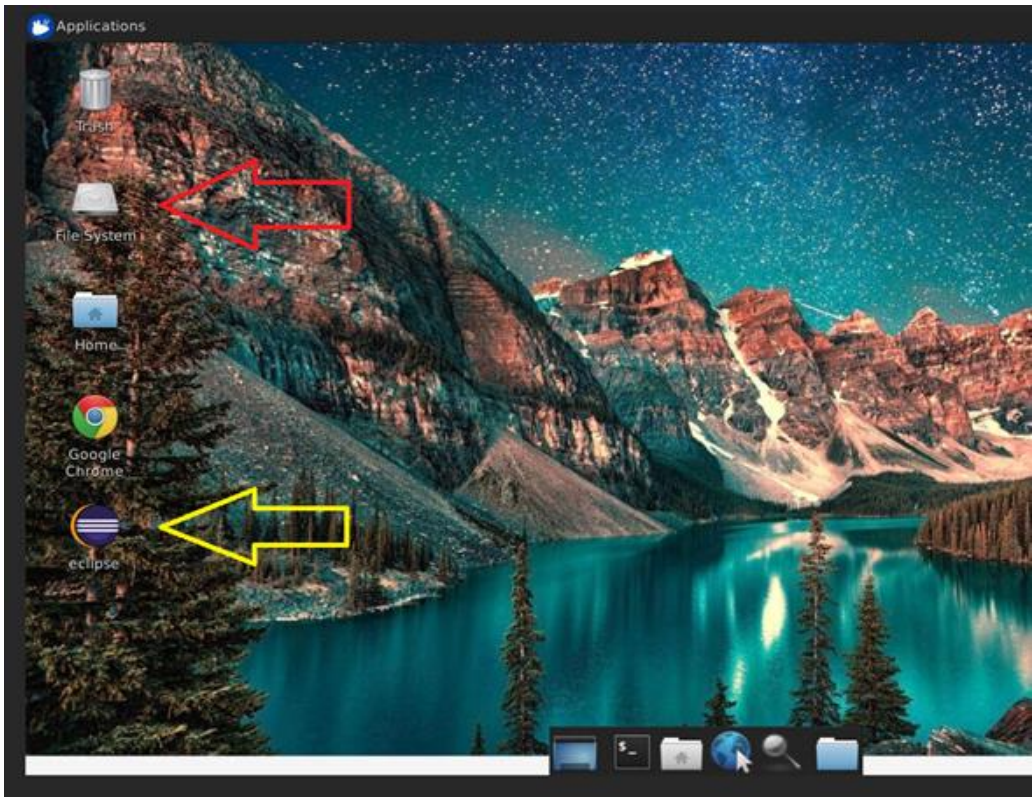


2. Click the **Virtual Desktop** link from the Virtual Desktop module in Brightspace. The Virtual Desktop should be accessed from Brightspace every time.

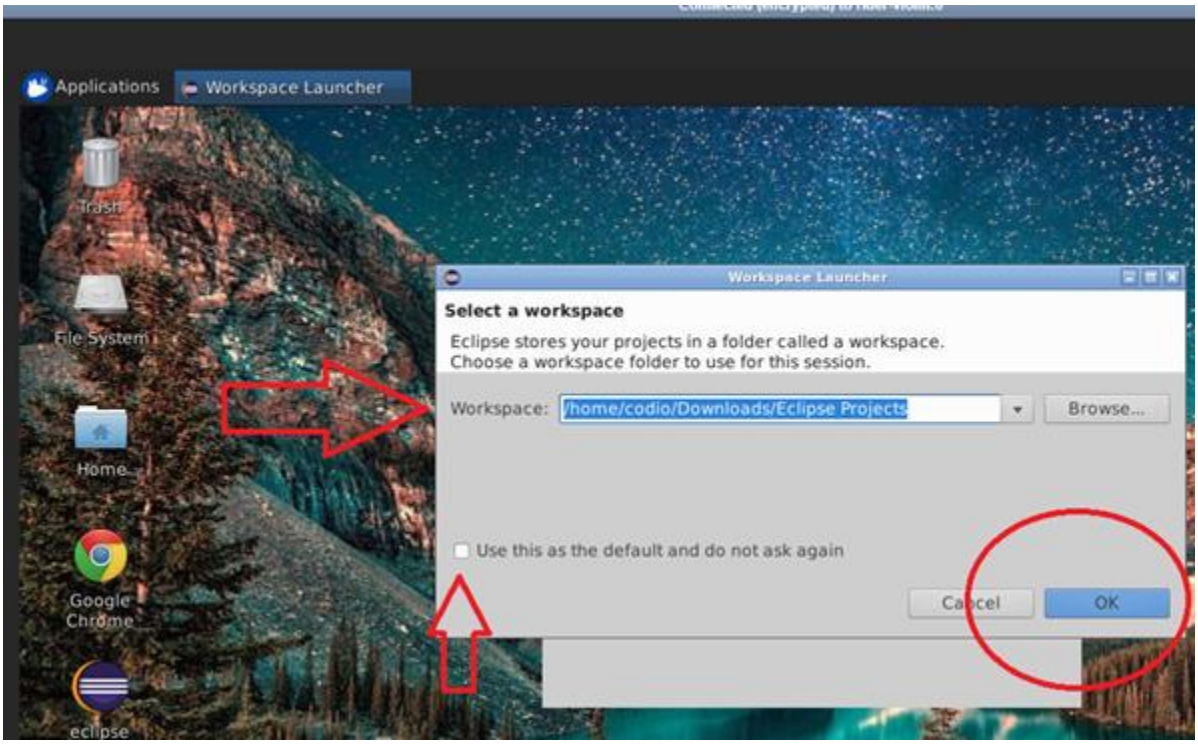


3. This will bring you directly to the virtual desktop in Codio.

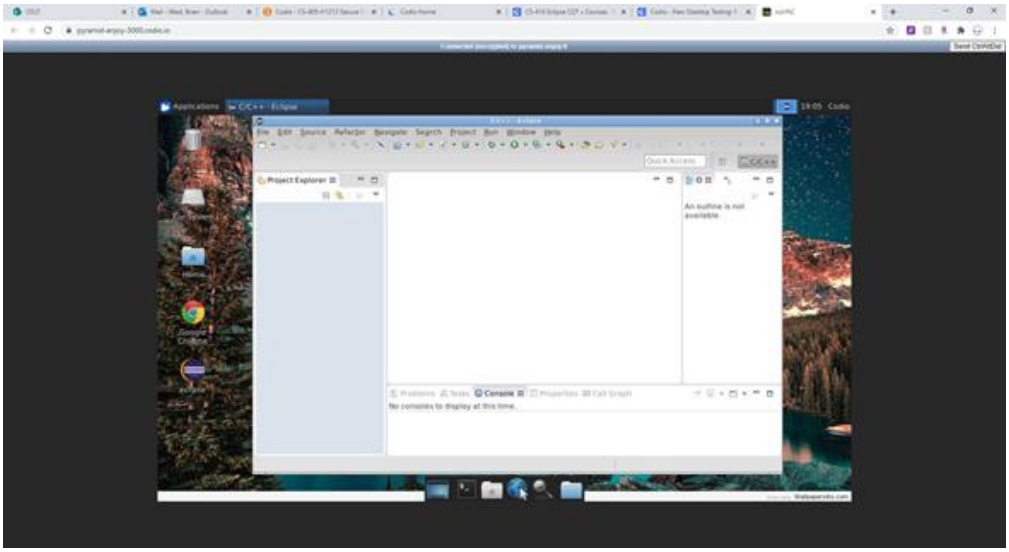
4. The virtual desktop will work the same as your desktop on your computer at home. As you can see, there are a number of icons for you to use. There are likely more than you see in the following image. You can access files that you have created through the File System icon (red arrow) any time you would like simply by double-clicking on it. For now, look at the Eclipse icon (yellow arrow). Double-click on that icon to access the Eclipse program.



5. A window should appear similar to the one in the following image. This allows you to select a workspace location. It is recommended that you use the default workspace. You can also click the **Use this as the default** check box if you do not want this to appear each time you open Eclipse. Now select the **OK** button.

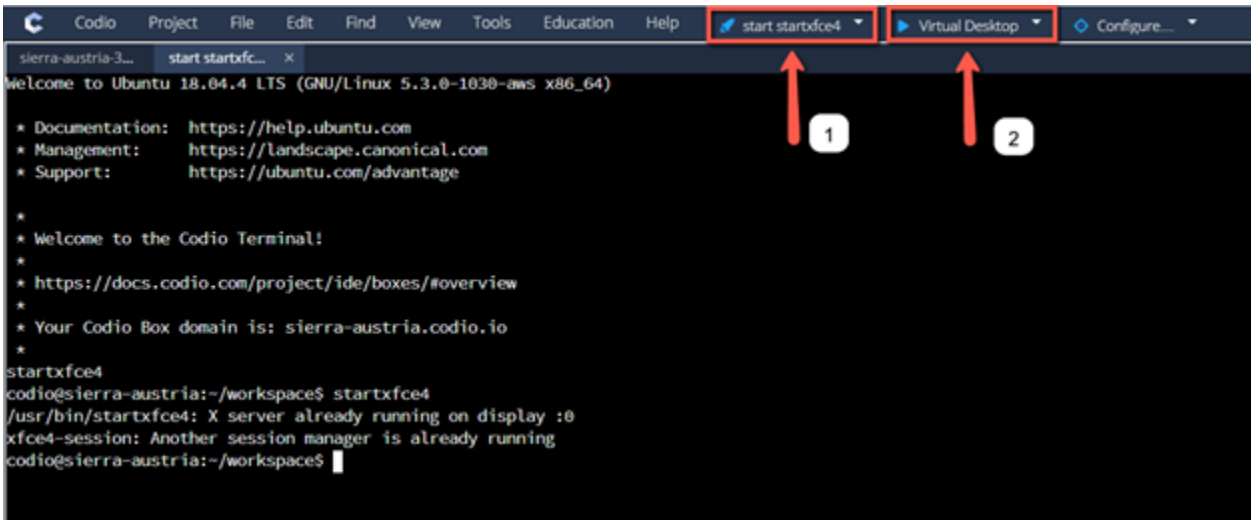


6. The Eclipse desktop has appeared. You are free to write your programs, test them, and run them.



Troubleshooting

1. If the Codio Virtual Desktop does not open automatically, click the **start startxfce4** button first, then the **Virtual Desktop** button to manually start the desktop.



Section 3: Helpful Hints

Working with Codio, or Unix in general, can be tricky. It's very important to be careful when dealing with code, especially if you don't have control over the actual machine (for instance, when you are accessing it remotely, as in Codio). Here are some recommendations to help you avoid common mistakes when working in such environments:

- Always make a backup of your files. If you're using code given to you in class by your instructor, make a copy of it and move it to a separate folder just in case you need to get back to it.

- Always keep your old code. Every time you modify your code, make a new version of the file. This way you can backtrack your work if you find out that you took the wrong path or wrote wrong code and want to revert to previous code.

- Stay organized; make a folder for each assignment.

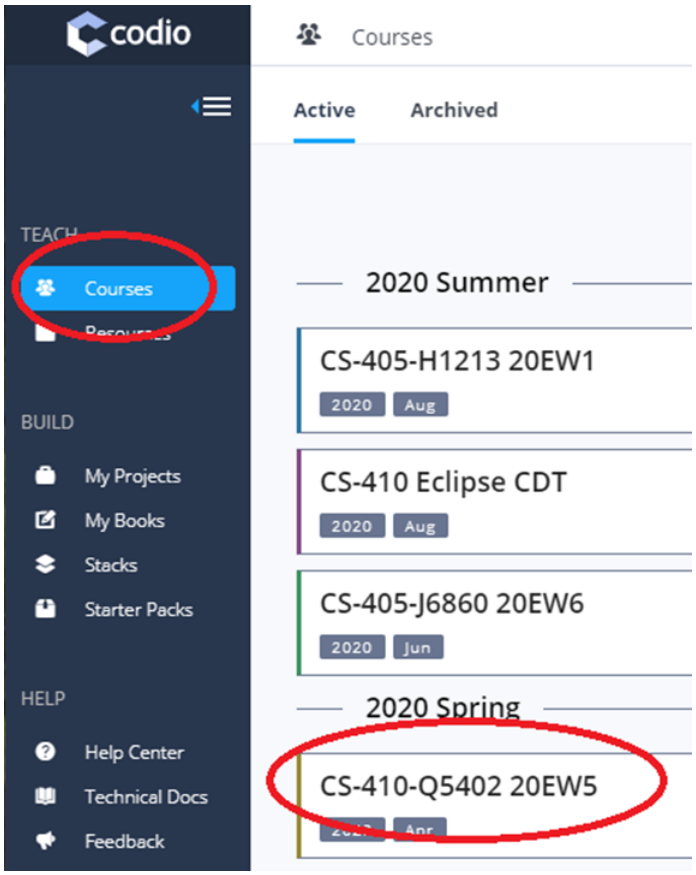
- When working with assembly, make sure your assembly is being built in the same environment as your class environment. Codio uses specific processors, and the class uses G++ to generate assembly. Assembly created by G++ differs from assembly created by ASM or other assemblers.
- Codio is used as a Linux box. Use the following Unix commands:

Command	Description	How to Use It
ls	Lists the files and directories in folder	ls
cp	Creates a duplicate/copy of a file	cp filename1 newfilename
mkdir	Creates a new folder	mkdir directoryname
mv	Renames a file or a directory	mv oldfilename newfilename mv olddirectoryname newdirectoryname
clear	Clears the screen	Clear
cd	Changes directory	cd directoryname
cd ..	Goes back one directory (up one level) Be careful: You need the space between the d and the first period (.)	cd ..

Section 4: Downloading and Uploading Files

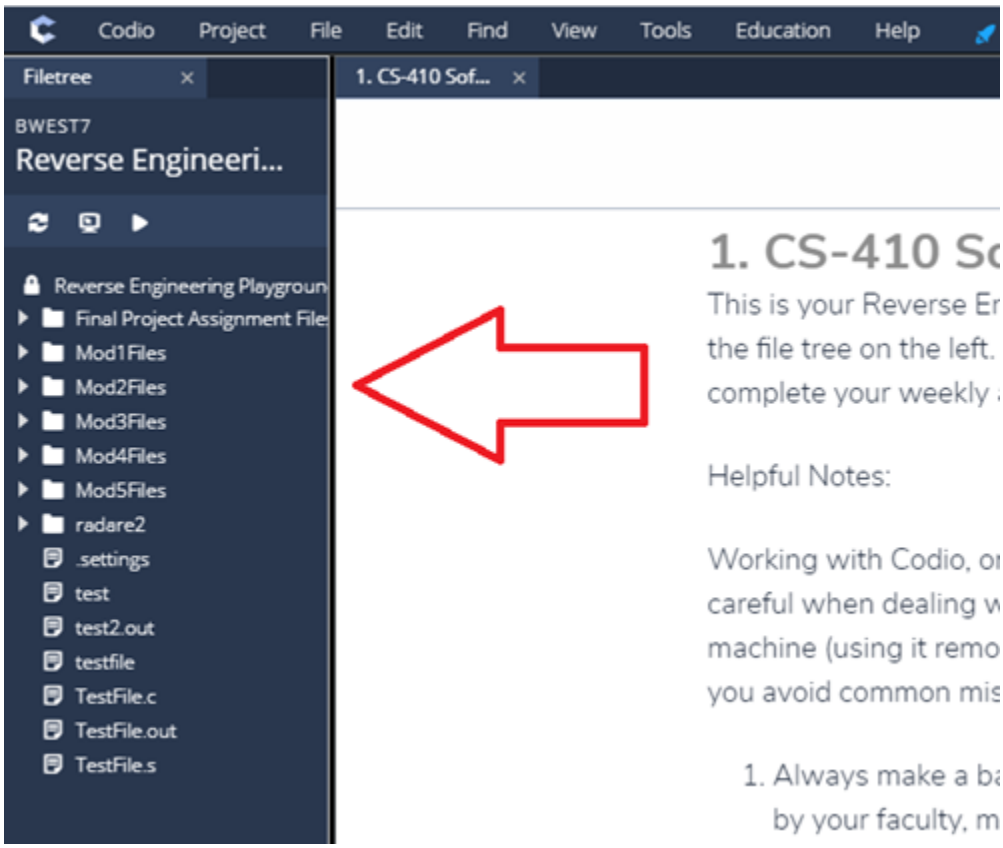
Downloading Files from Codio

1. Log in to Codio. If the “Courses” option is not selected, please select it. Then, select your course.

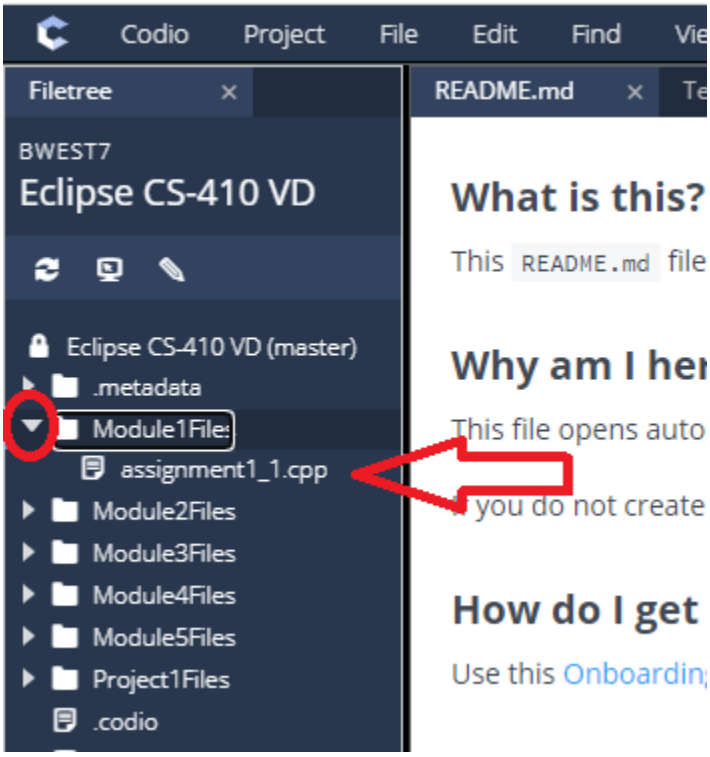


2. Select the appropriate “playground” for your course.

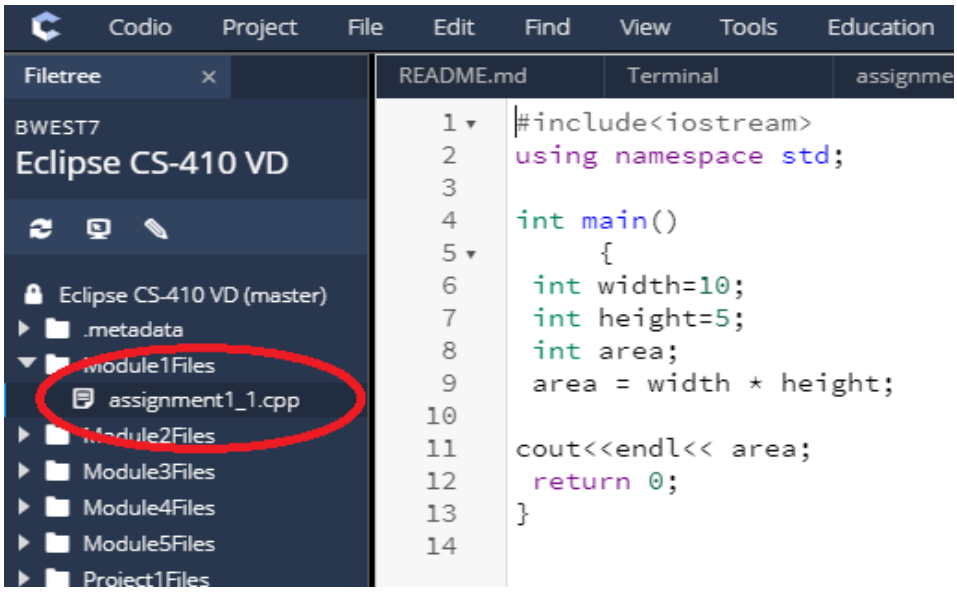
3. The course folders are displayed in the left frame, also referred to as the file tree within Codio.



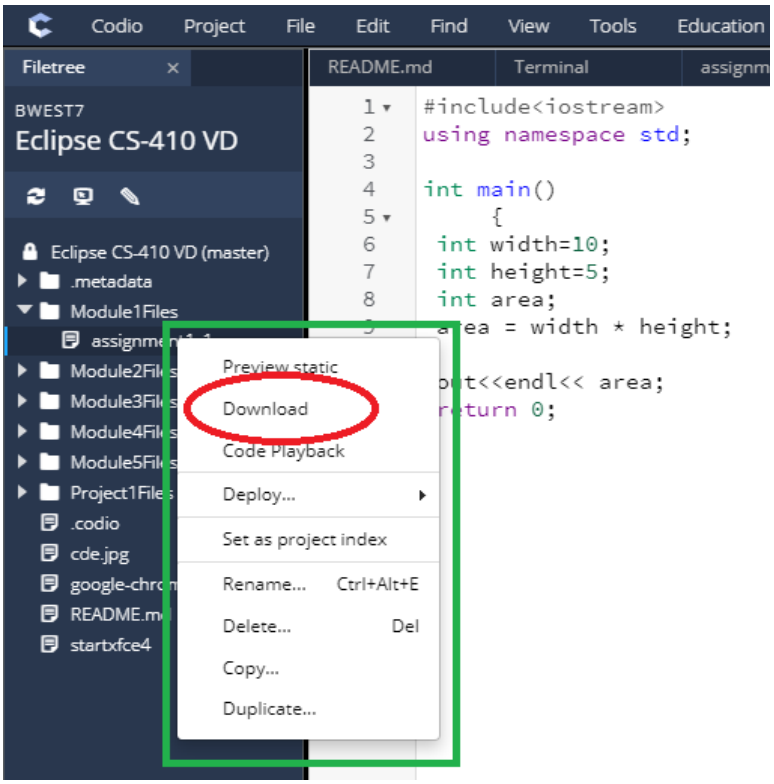
4. Select the carrot next to the weekly folder and its contents will appear indented beneath it.



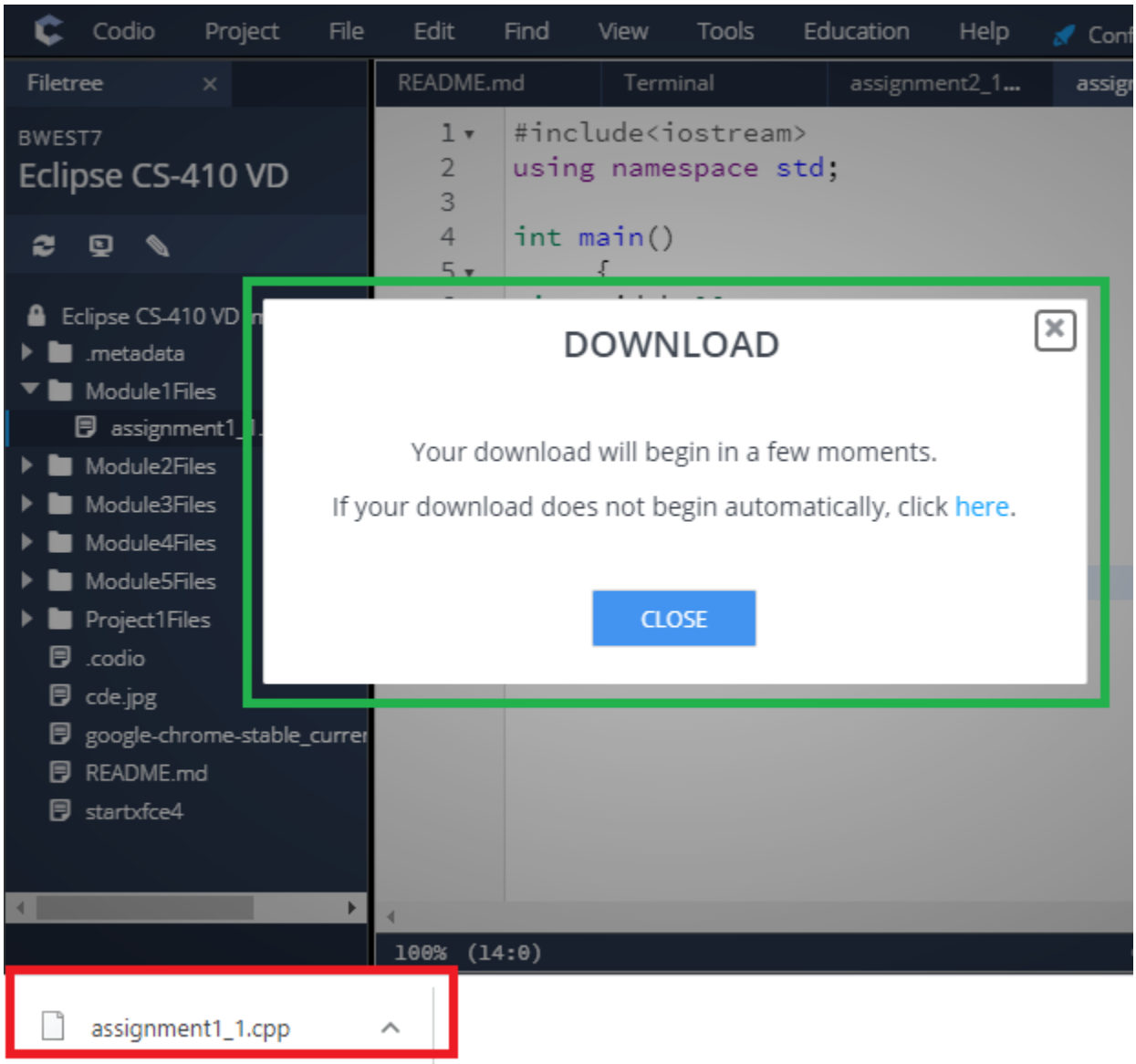
5. For example, if you select the CPP file, its contents will be displayed in the code window to the right.



6. Right-click the code file and a shortcut window will appear. Select the **Download** option on that menu.

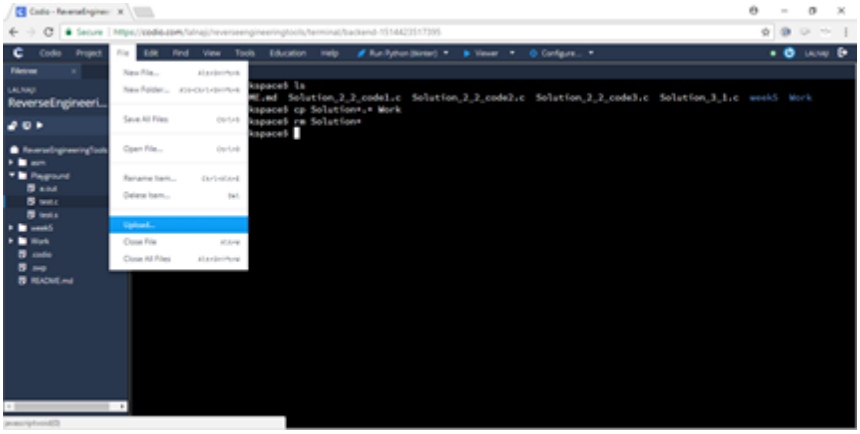


7. This will likely bring up a Download message. If you look at the bottom left of the window, you should see the assignment name. This indicates that it was downloaded to the Downloads folder, unless you have previously changed this default on your computer.



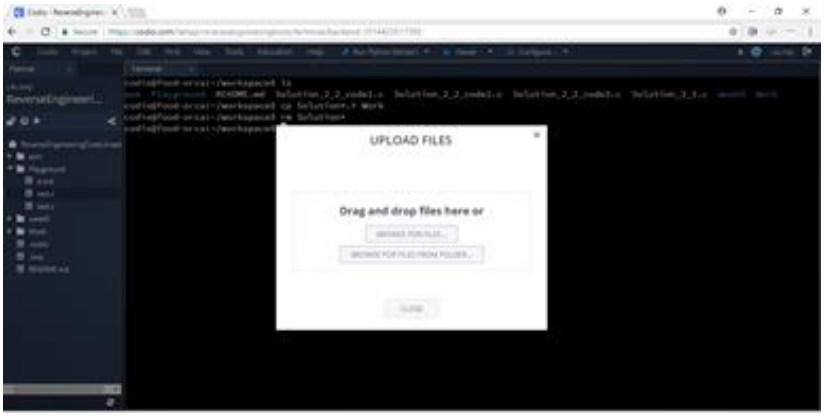
Uploading Files to Codio

1. Select the File menu in the top navigation bar in Codio. Then select **Upload**.



2. Either drag the file you want to upload into the pop-up window titled Upload Files, or click the **Browse for Files** button and select your file.

3. Once you click **OK**, the file will be uploaded to the root folder.



Let's assume the file you uploaded is called test.cpp. To move that file into a folder named WORKAREA, you would type the following command:

mv test.cpp WORKAREA.

Section 5: Using the Bless Hex Editor

Bless is a hex editor application that runs on Unix, was written by Ztrela Knowledge Solutions Pvt. Ltd, and can be run within Codio. The application runs on Linux, but the authors claim it can run on other platforms as well.

Some of the features and advantages of Bless listed on the Ztrela Knowledge Solutions website include the following:

- Efficient editing of large data files
- Multilevel undo/redo operations

- Customizable data views
- Fast data rendering on screen
- Multiple tabs
- Fast find and replace operations
- Conversion table
- Advanced copy/paste capabilities
- Multithreaded search and save operations
- Export to text and html (others with plugins)
- Extensibility with plugins

Learn more about how to use the Bless hex editor to understand binary codes by referring to the Hex Editing: First Steps.

Section 6: Compiling Assembly From Binary

The command g++ can compile assembly code. You can create an executable file using assembly as an input file. Let's use the Hello.s file created in the previous appendix as an example. To compile Hello.s, type the following:

g++ -Hello.s -o Hello.o

To run the binary file, type this:

./Hello.o

This should print the following:

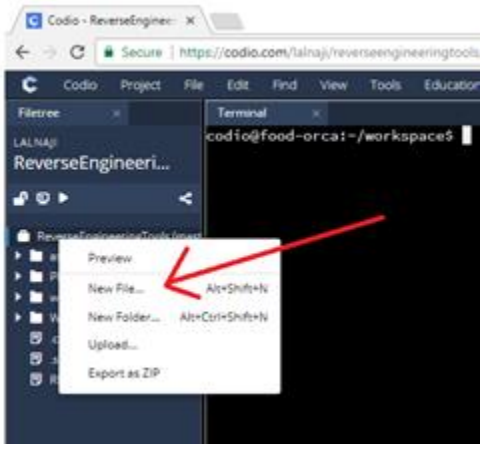
Welcome to my world

Section 7: Creating a Simple C++ Program Using the G++ Compiler

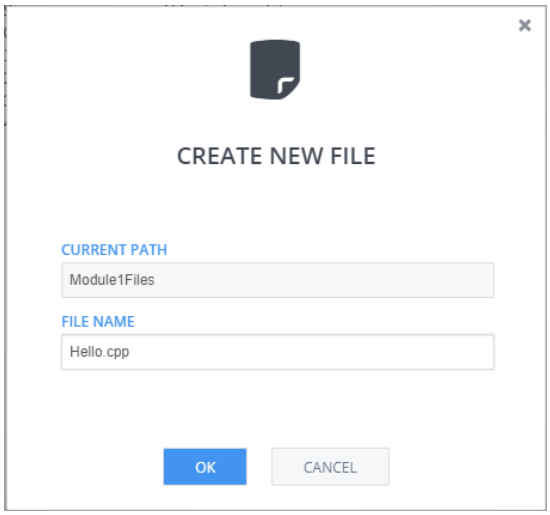
Here is a walkthrough on writing in C++ using the G++ compiler. Follow these steps to create a simple C++ program in the Codio environment.

Creating a New File in Codio

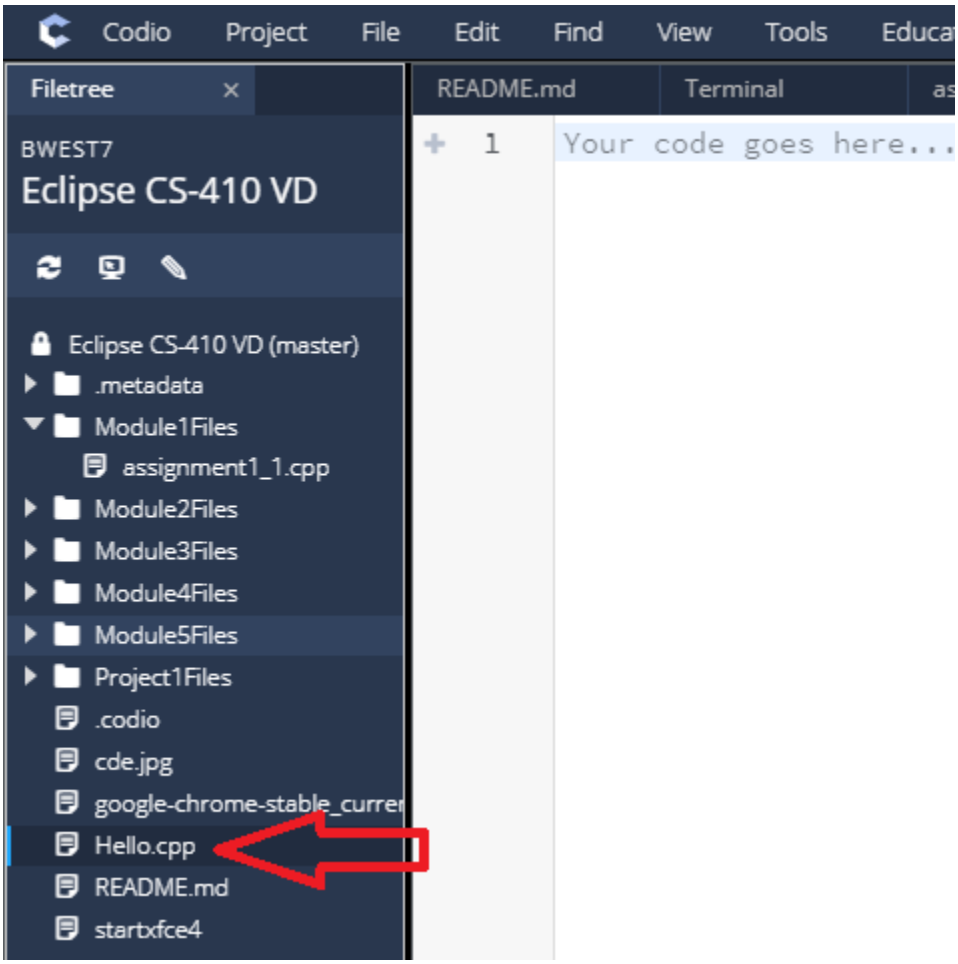
4. First you need to create a new file. Right-click the Codio project/environment and then select **New File**. You could also click the File link at the top and select **New File** from the drop-down menu.



5. In the pop-up window, enter the file name (for example, “hello.cpp”).



6. After you click **OK**, a file will be created in your directory. You can see the file name in the left panel. If you click **Hello.cpp** in the left file-tree panel, the program will open the file for you to write/edit.



7. Now write the code displayed on the right in your file and then save it.

```
#include <iostream>

 using namespace std;
int main()
{

 cout << "Welcome to my world" << endl;
 return 0;
}
```

8. To save the file, you can hold CTRL and type S, or you can select **Save All** from the File menu.

That's it. You just created your first C++ program in Unix! To compile the file and run it, see the next section, Compiling C++ Code.

Section 8: Compiling C++ Code

Here is a walkthrough on compiling using G++. To compile the C++ code created from the Writing C++ Code section above, you will use the g++ command. When compiling C++ code, you need to specify the following:

- Input file name: This is the file that contains your C++ code.
- Use the argument **-o** to indicate you want to create an output file.
- The output file name: The output file name must have a different extension (or name) than the input file name. For example, to compile Hello.cpp, we can create a Hello.out as an output file.

Let's try it. After creating Hello.cpp as explained the previous section, type the following in your environment:

g++ Hello.cpp -o Hello.out

Notice that this command will create a file in your folder called Hello.out. To run the executable, type this:

./Hello.out

That's it! You just compiled your first C++ program. Congratulations!