CSCI 1300/1310 - Intro to Computer Programming
Instructor: Hoenigman/Knox
Recitation 2

Due Date: Sunday, Jan. 29th, by 5pm

Hello, World

The first program that we usually write in any language we're learning is Hello, World. A Hello, World program just prints "Hello, World" to the screen. CodeBlocks is the development environment we're going to use for C++ in this class.

Step 1: Open CodeBlocks

Click on the CodeBlocks icon in the VM just above the Geany icon (**Figure 1**).

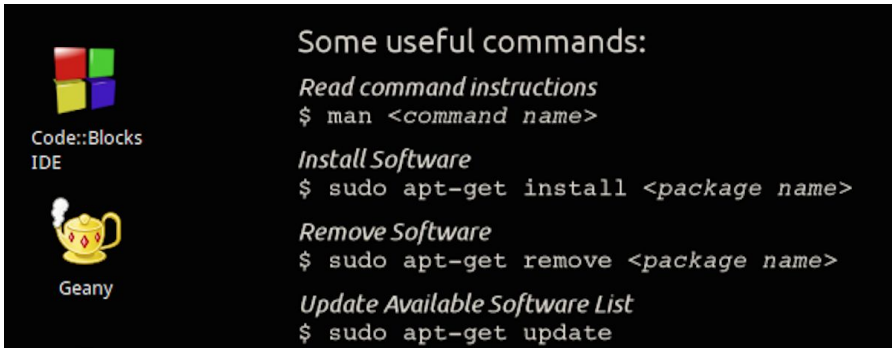
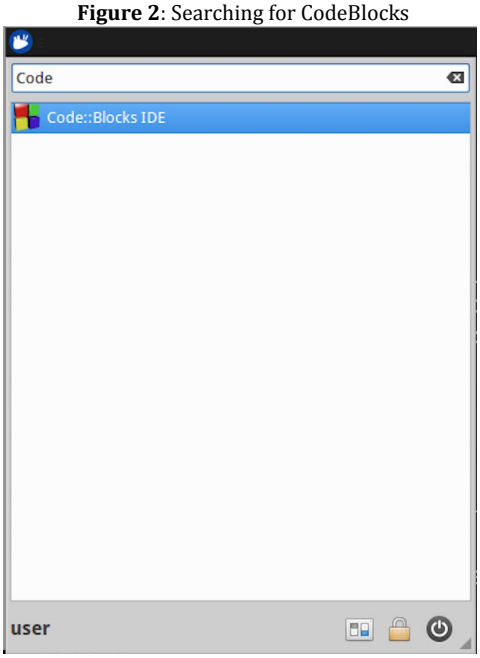


Figure 1: VirtualBox Desktop

Figure 2: Searching for CodeBlocks



Troubleshooting: Cannot find CodeBlocks icon
If the icon doesn't exist, you can also load the program by searching for it. Click on the blue and white icon in the upper left corner of the VM. A drop-down menu will appear with a textbox where you can type the name of the program you want to search for. In the textbox, type Codeblocks. You will only need to type a portion of the word and the CodeBlocks program should show up in a list of programs that match your search. Click on the CodeBlocks program (**Figure 2**).

When CodeBlocks loads, you will be presented with the following screen (**Figure 3**) where you can create a new project, open an existing project, as well as a few other options. You can ignore this screen for now. In future classes, we will be creating projects, but not yet.

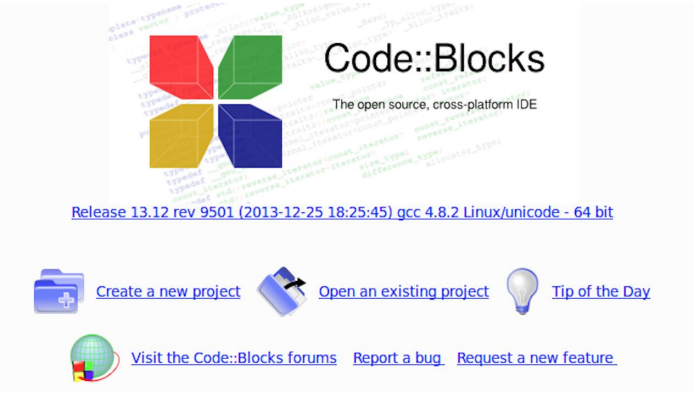


Figure 3: CodeBlocks Welcome Screen

Step 2: Open an Empty File

Select File -> New -> Empty File. You can also click on the New File icon under the File menu. A new, blank file called Untitled1 will be opened. (**Figure 3**)

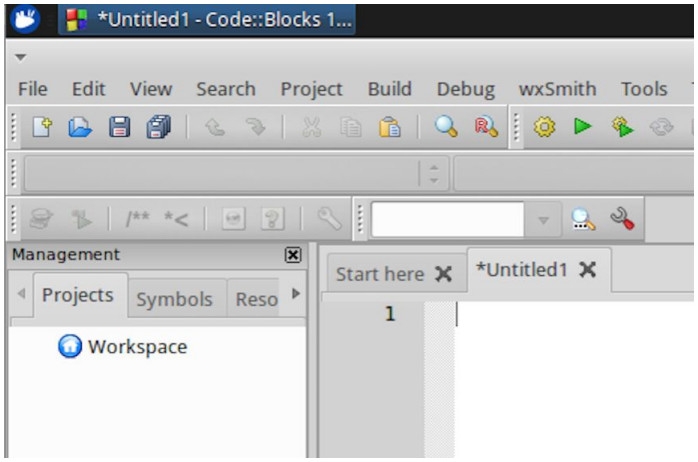


Figure 3: Empty File in CodeBlocks

Step 3: Your First Code!

Starting on line 1 in Untitled1, type the following code:

```
1   #include <iostream>
2
3   using namespace std;
4
5   int main(){
6
7       cout<<"Hello, World";
8
9   }
10
```

Step 4: Saving Your File

Save the file. Name it main.cpp. The .cpp extension on the filename tells CodeBlocks that the file should be read as C++ code. Once you save it, the lines in the file should be color coded to reflect what they do in the program. This is called *syntax highlighting* (**Figure 4**).

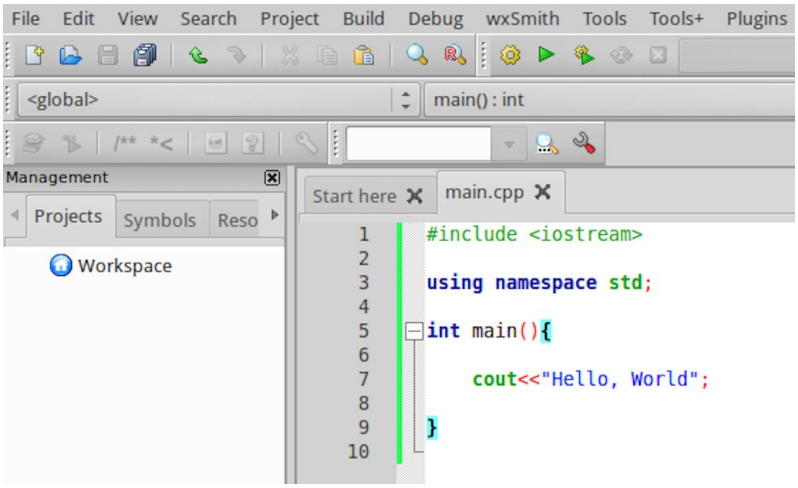
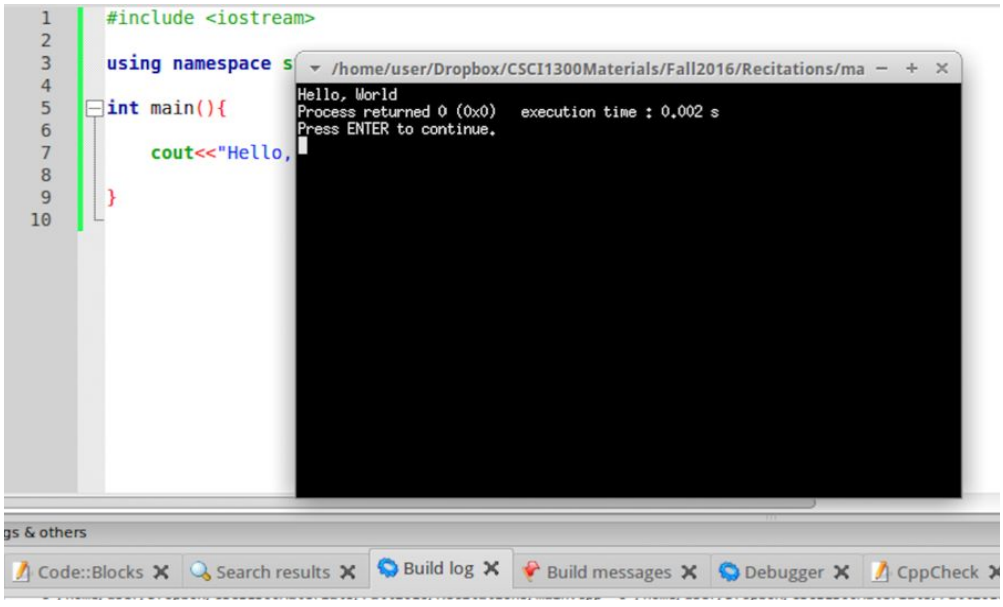


Figure 4: Syntax Highlighting

Important: You should save your work <u>frequently</u> in CodeBlocks to avoid losing your work in the event of the program crashing.

Step 5: Running Your Code

To run the program, click on the icon with the green arrow and yellow gears just under the Tools menu. If it works, you should see a terminal window load and print Hello, World.



Step 6: Update Your Code

Now, change the cout statement so that your program prints out the following:

Hello, CS1300 World!

Step 7: Submitting Your Code on COG

Throughout the semester we will be using the autograder, COG. The score that you receive on COG will automatically be sent to your Moodle gradebook and logged as your grade. However, you may upload your assignment to COG as many times as you want to. This enables you to maximize your score. To acclimate you to using COG, you will be uploading your recitation 2 assignment to COG.

Step A: Rename .cpp File

First, you must appropriately name your .cpp file. In your project you named the file main.cpp. However, in order for COG to recognize your file you must give it a specific name. Locate your main.cpp file and rename it as follows:

lastname_firstname_recitation2.cpp

For example, if your name is John Smith your file should be named:

smith_john_recitation2.cpp

Step B: Compress Your .cpp File to .zip

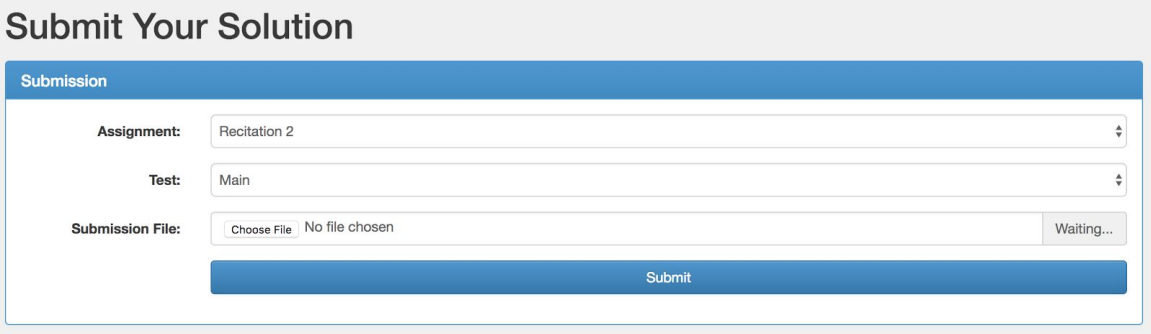
Second, you must compress your renamed .cpp file into a .zip file. Locate your .cpp file, right click it, and chose the option to compress it.

Note: You *must* compress it as a .zip file - no other compressed formats are permitted.

Note: You can name you .zip file anything you want.

Step C: Upload .zip to COG Website

Next, navigate to https://web-cog-csci1300.cs.colorado.edu. You will then select 'Recitation 2' from the assignment drop-down menu. Next to 'Submission File', select the 'Choose File' option. This will open up a new window. Navigate to the location of your .zip file from step B and choose it for upload. Lastly, click the large blue 'Submit' button.



Step D: Reading the COG Output

After you upload your .zip file and submit it, COG will compile your code and check its output against an expected output. This process will be output to the dialogue box that appears after pressing 'Submit.'

Important: If your code does not compile, you will receive a **ZERO**.

The dialogue box will give you insight into what parts of your code are incorrect and what the correct output should have been. For example, if you did not change the program from the original cout of "Hello, World", the program will take off 50% of the score and it will notify you of the expected cout.

Important: You must match the cout exactly as specified in the assignments for COG to award you credit.

Lastly, COG will tell you the score you received on the assignment. If you are not satisfied with this score, follow the advice in the COG output to debug your program. That is, rewrite it, compress it into a .zip file, and reupload it. You can do this as many times as you wish.

Step 8: Submitting Your Code to Moodle

In addition to submitting your code to COG, you **must** submit your files to Moodle. Utilize COG to finalize your assignment and maximize your score. Once you are satisfied with your grade, you must submit the files to Moodle.

Navigate to the correct week on Moodle and look for the following link under the Recitation section. Use this link to submit your **.cpp** file.