CSCI 2270 – Data Structures and Algorithms

Instructor: Hoenigman/Zagrodzki/Zietz

Recitation 1

Objectives:

1. Setup your Moodle account and test that it works.
2. Install Sublime Text and build files.
3. Programming exercise to learn how to build on Sublime Text.

Table of Contents:

Moodle Account

All students in CSCI 2270 this semester will be accessing course materials through the Computer Science Moodle:

http://moodle.cs.colorado.edu

To use Moodle, you need to login using your CU identikey and password, and then enrol in your class. Once you've logged in, select the CSCI2270-S18 – Hoenigman class and enter the enrolment key: csci2270 to enrol. Once you're enrolled in the class, you should see the course materials that have been uploaded so far, organized by weeks.

Moodle Account Test

To test that you have set up your Moodle account and are able to submit your assignments, **upload a file to the Recitation 0 Submit link**. It doesn't matter what file you submit; this exercise is to verify that you know how to upload your assignments before the first assignment is due. **You should receive a confirmation email once your file is submitted**. Verify that you receive that email. If you do not receive this email, you did not submit correctly.

Installing Sublime Text:

We will be using Sublime Text as the default code editor for this course. We will be teaching you how to install sublime text today and how to build a file with it. Here are the instructions to install it:

1. Go to this link.
2. Download the file pertaining to your OS.
3. Based on your OS the file extension will differ, install the file in the following manner:
   - Windows Installation (Normal Version NOT Portable Version):
     a) Download the installer
     b) Double click on the installer
   - MacOS:
     a) Download .dmg file
     b) Open .dmg file
     c) Drag the Sublime Text 3 bundle into the Applications folder
     d) To create a symbolic link to use at the command line issue the following command at the terminal:

```
ln -s  "/Applications/Sublime Text.app/Contents/SharedSupport/bin/subl" /usr/local/bin/subl
```

Note: Symbolic links or symlinks are files that simply contain the path of the target file or directory stored as text. If you move the target file, the symlink will break because it still points to the original location in the link. Thus, while using the link in the command line we will have to use a shorter and simpler link.

   - Linux:
     a) You can download the package and uncompress it manually. Alternatively, you can use the command line.
     b) For Ubuntu:

```
cd ~
wget http://c758482.r82.cf2.rackcdn.com/sublime-text_build-3083_amd64.deb
```

     c) For Other Linux Distributions:

```
cd ~
wget http://c758482.r82.cf2.rackcdn.com/sublime_text_3_build_3083_x64.tar.bz2
```

d) Now we should move the uncompressed files to an appropriate location.

```
sudo mv Sublime\ Text\ 3 /opt/
```

Building the files:

Note: Packages are a collection of resource files used by Sublime Text: plugins, syntax highlighting definitions, menus, snippets and more. Sublime Text ships with several packages, and more user created ones are available. We will use packages to build and run C++ files from within sublime text.

Using the Package Controller we can automatically install and update packages. Thus, we will be using the Package Controller to help us setup the needed packages.

We will be using "PackageResourceViewer" to enable us to view and edit package resources in Sublime Text.

1. Install PackageResourceViewer plugin
    a) Installation through package control is recommended. It will handle updating your packages as they become available.
    b) The console is accessed via the ctrl+` shortcut or the View > Show Console menu. Once open, paste the appropriate Python code for your version of Sublime Text into the console from this link.
    c) To install, do the following.
    d) In the Command Palette, enter Package Control: Install Package
    e) Search for PackageResourceViewer.

2. **If the method mentioned above does not work follow this:**
    - Download from this link into the packages directory. Ensure it is placed in a folder named **"PackageResourceViewer"**. By default, the Packages directories for Sublime Text 2 are located at:
        - OS X: ~/Library/Application Support/Sublime Text 2/Packages/
        - Windows: %APPDATA%\Sublime Text 2\Packages\
        - Linux: ~/.config/sublime-text-2/Packages/
    - By default, the Packages directories for Sublime Text 3 are located at:
        - OS X: ~/Library/Application Support/Sublime Text 3/Packages/

- Windows: %APPDATA%\Sublime Text 3\Packages\
- Linux: ~/.config/sublime-text-3/Packages/

3. Open the command center by entering Ctrl + Shift + p
4. Type “PackageResourceviewer: Open Resource”
5. Then C++ –> C++ Single File.Sublime-build , this will open the build config file for C++
6. Where ever you g++ add **-std=c++11** infront of it and save the file.

```
{
	"shell_cmd": "g++ -std=c++11 \"${file}\" -o \"${file_path}/${file_base_name}\"",
	"file_regex": "^(..[^:]*):([0-9]+):?([0-9]+)?:? (.*)$",
	"working_dir": "${file_path}",
	"selector": "source.c, source.c++",
	"variants":
	[
		{
			"name": "Run",
			"shell_cmd": "g++ -std=c++11 \"${file}\" -o \"${file_path}/${file_base_name}\"
&& \"${file_path}/${file_base_name}\""
		}
	]
}
```

7. Save it. Rename the file as “MyC++Build” or anything you want and save it.
8. Go into Tools > Build System and then Select your Build “MyC++Build”.
9. Later open any folder using Ctrl+O where you usually store your programs. In that folder create a file name **“input.txt”.**

Now start coding. Write a program and then save it in that folder only. The benefit of doing is that whenever you have to give inputs just open the created input.txt enter your input data and save it. Later perform Ctrl+B for compiling and Ctrl+Shift+B for Run.

Recitation Exercise

We will learn how to use Sublime Text to create a single C++ file.

This is a hello world program in C++.

helloWorld.cpp:

```
// my first csci2270 program in C++
#include <iostream>

int main()
{
  std::cout << "Hello World!";
}
```

Now save this file as a .cpp file.

In the next Recitation we will also learn how to use Sublime Text to build C++ files and run them from within Sublime Text.