**ECE 1111: Engineering Computation I**

**Laboratory No. 1: Infrastructure**

**Goal:** The goal of this lab is to make sure you have set up the necessary infrastructure to function efficiently in this course. The topics covered in this lab were distributed via email prior to the start of the course as pre-course assignments intended to position you for success in this course. These assignments introduce you to the basic computing environment we will use throughout this course.

**Deliverables:**

1. *Infrastructure:* Demonstrate that you completed the pre-class assignments. This includes:
2. You can successfully log into the class server, ece-000.
3. Using emacs by invoking the command isip\_e, demonstrate that you can open a text file on ece-000, edit it, save it to a new file, and navigate the file using keyboard commands (e.g., forward one character, backward one character, forward one word, backward one word, cut and paste).
4. Demonstrate that you can navigate directories from the command line, search for previous commands using emacs-like bindings, copy files, delete files, and move files. All of this must be done from a terminal window using the Linux command line.
5. Demonstrate that you have installed VS Code and co-Pilot on your laptop. See this URL for videos on this:

https://isip.piconepress.com/courses/temple/ece\_1111/resources/tutorials/tips\_vscode/

Demonstrate that you can open this source code file on ece-000:

/data/courses/ece\_1111/current/labs/lab\_01/picone\_joseph/example.cc

Edit the file and save it to your lab directory:

/data/courses/ece\_1111/current/labs/lab\_01/<lastname\_firstname>

Verify from the terminal window that your edited file resides in your personal lab directory by doing a directory listing. Demonstrate the use of co-Pilot by looking up any of the syntax in your source code file that you don’t understand.

1. *Google Search:* Demonstrate that you can Google search for a Python program that prints “hello world” to the terminal screen (stdout). Copy this program to:

/data/courses/ece\_1111/current/labs/lab\_01/<lastname\_firstname>

with a filename of *myprog.py*. Run it using the Python interpreter (“*python myprog.py*”).

**The Check Off Process:**

For each lab, you will meet with a lab TA during the lab session. The TA will ask you to demonstrate your solutions and will ask you to answer some questions about your code. Though we encourage group work in this course, your ability to answer questions about your code will be a large part of your lab grade. Your TA will form an impression of how well you have mastered the concepts in this lab by the way you answer questions about your solutions and assign a grade accordingly. If you fail to provide working commands or programs, you will receive a score no higher than 50/100 ­­– quite possibly much lower.

Note you only get one chance to check off, so be well-practiced before you attempt the check off.

**Description:**

As explained prior to the start of class, getting a head start on learning some basic computing skills in this class is important. It takes time for these concepts to sink in, and we can’t let that get in the way of the main topic of this course – learning how to design and write computer programs. Completion of these assignments will help you start the class on a strong note. Learning how to navigate the command line and edit files in emacs are extremely important skills in this class.

**Summary:**

The first program we often write when learning a new programming language is a program that prints “hello world” to a terminal window. Though we often joke about this, it is, in fact, one of the first programs you write in a new language because input/output (I/O) is extremely important in engineering. If you understand how to do I/O in a programming language, you understand a great deal about how a language works. Engineers move data in and out of programs constantly. We deal with real-world signals and systems and often process physical signals such as speech, heart rate, blood pressure, video, etc.

By now, you should by now be familiar with the basics of simple text editor like emacs, so creating and saving a file should be relatively easy for you. The most basic way to develop code is to use a text editor such as emacs and a command line debugger such as gdb (which we will cover later in the course). More advanced software development makes use of an integrated development environment (IDE) such as VSCode that makes some of this process much easier.

In this lab, you have learned the basics of how to log into the AWS server, navigate the filesystem and run both a text editor and an IDE.