

# Project 5.5 Assignment

## I VIDEOS TO WATCH

Links to videos that you may find useful

- CLI and Python basics

This shows some of file access basics and terminal access. As long as you are able to access a shell or terminal and launch python with command line options, then you should be fine. See the next section for the *FastX* links. I have requested that each of you have access. The GNOME interface you will see is very slightly different, but the general topics are the same.

- Object Oriented Text Processing

Probably the most relevant item for getting started doing things most related to this activity.

- Directory Access

Another look at directory access and going through all files in a directory from within a program. This is typically a confusing item students.

## 2 GETTING A LINUX GUI GOING ... IF YOU WANT

This piece is presented in class. You are able to access a Linux GUI through a web browser. This is helpful to test to ensure that our code will work on both your own machine (which may be Windows or Mac based) as well as another POSIX environment like a Linux filesystem.

We will be doing some basic processing with the CLI. For now, let's try it on a Linux machine. You can figure out how to get VSCode setup to work for you with a command line interface later.

We will use the FastX web-based interface. Bucknell has some information at this link: [Bucknell LIT Info](#)

Or just use one of these:

- <https://linuxremote1.bucknell.edu:3300>
- <https://linuxremote2.bucknell.edu:3300>
- <https://linuxremote3.bucknell.edu:3300>

## 3 ASSIGNMENT

bunny

Your goal is to call your program and ask have it count the number of times “bunny” occurs in a text file. The file to be scanned will be passed into your program from the command line. Your program should print out some helpful information to the output, and then write the number of times “bunny” occurs into a separate file. There are two parts to this—scanning a single file, and then scanning through *all files* in a directory.

### 3.1 PART I—SINGLE FILE

Run your bunny-scanning program with the `FirstFile.txt` and then `SecondFile.txt` and look at the outputs. Do they match with my output or what you expected it to be?

---

```
$> python bunny.py FirstFile.txt
In FirstFile.txt I counted bunny 1 times.
```

---

Once this matches, then your program should write the number `1` to the file `FirstFile.out`. After running your program, printing out the contents of `FirstFile.out` (I am using the unix command `cat` to display the file contents) should look like below.

---

```
$> cat FirstFile.out
1
```

---

The `cat` command will output the contents of a file to the command line. Do your results match?

---

```
$> python bunny.py SecondFile.txt
In SecondFile.txt I counted bunny 4 times.
```

---

The number `4` should then be written to `SecondFile.out`.

## 4 SECOND PART

Now we will pass a directory to the bunny-counting program. When we pass a directory, for *each file inside the directory* that ends in `.txt`, you should look for “bunny” in the file. We do not need to worry about recursively descending through all the directories. Print out the number of times that “bunny” is in *all* the `.txt` files in the directory and then write your results to a `.out` file that is *inside the directory* with the name `<directory name>.out`, where you replace `<directory name>` with the name of the directory you scanned. An example will help this make more sense.

---

```
$> python bunny.py ./hare/
In ./hare/ I counted bunny 10 times.
```

---

The number `10` should then be stored in a file named `./hare/hare.out`.

---

```
$> cat ./hare/hare.out
10
```

---

## 5 FILE LOCATIONS

You can download the files at <https://eg.bucknell.edu/~sjt015/eceg431/p55Items/p55Assignment/project5-5.zip> or this link: Project 5-5 zip file.

## 6 ZIP FILE CONTENTS

The zip file should could contain the following files: