

## CS 360 Review Exam 1 - License, Linux, Git, GitHub

### License

Explain why you would use the GPL versus just putting your code in the public domain when you start an open source project.

What are the 4 freedoms?

What are the basic protections of copyright?

Many people (or a few vocal people) started out this course with a negative view of copyright law. If there was no copyright law, how would open source software be different?

True or False. Defend your answer: Any code posted on GitHub is open source so Sally can use this code as she wishes.

### Linux

**Review the Linux exercise in class.**

Display the contents of LICENSE.txt on the screen.

You started **nano** yesterday. Write a command that you can run from the command line to check to see if **nano** is still running.

You are currently in the directory CS360.

```
punetid@bart:~/CS360>
```

Write the full command you would use to find all files in CS360 and in all the subdirectories whose filename ends in **.txt** and find each occurrence of the word **GNU** in those files.

### Version Control

Give two reasons you might use a version control system on a project.

### Git / GitHub

How are the commands 'git init' and 'git clone' different? Explain an instance when you would need to use each command.

Give a series of commits where the use of --no-ff has no effect.

Give a series of commits where the use of --no-ff has an effect.

Explain why you should work on a branch in your local git repository rather than on master.

How is git pull different than git fetch?

What does **origin** refer to? Don't just tell me the address **origin** points to. Tell me why **origin** points to that address.

What does **upstream** refer to?

Suppose you ran the following command in your repository and it completes successfully.

```
chadd@bart:~> git push origin PUNETID_bug_1138
```

What does the string **PUNETID\_bug\_1138** refer to?

Why did you just run that command? What were you trying to accomplish?

Describe in detail what is happening in each of the following git commands. Assume you have forked a repository called **TestRepos** in GitHub and cloned that repository to your local hard drive

```
git checkout -b chaddcw-master master
```

```
git pull git@github.com:chaddcw/TestRepos.git master
```

```
git checkout master
```

```
git merge --no-ff chaddcw-master
```

```
git push origin master
```

Why is a pull request called *pull* and not called push?