I have no idea what I’m doing!
I have no idea what I’m doing!
Why is processing a sorted array faster than an unsorted array?

Here is a piece of C++ code that seems very peculiar. For some reason, it miraculously makes the code almost six times faster. #include <algorithm>

Edit an incorrect commit message in Git

I wrote the wrong thing in a commit message. How can I change the message if it has already been pushed yet.

How do you undo the last commit?

I committed a directory containing .class files instead of a directory containing .java files.
Edit an incorrect commit message in Git

I wrote the wrong thing in a commit message. How can I change the message? The commit has not been pushed yet.

How do you undo the last commit?

I committed a directory containing .class files instead of a directory containing .java files to Git. How can I undo this commit?

What is the correct JSON content type?

I've been messing around with JSON for some time, just pushing it out as text and it hasn't hurt anybody (that I know of), but I'd like to start doing things properly. I have seen so many purported ...

Delete a Git branch both locally and remotely

I want to delete a branch both locally and on my remote project fork on GitLab. Successfully Deleted Local, but now the remote still has the branch. How do I remove it there?
What is the correct JSON content type?

I've been messing around with JSON for some time, just pushing it out there as raw data (that I know of), but I'd like to start doing things properly. I have seen

```
json
content-type
```

Delete a Git branch both locally and remotely

I want to delete a branch both locally and on my remote project fork. I tried:

```
Branch $ git branch -D bugfix
Deleted branch bugfix (was 2a14ef7).
```

```
git
github
git-branch
git-remote
```
WHY DO WE CARE
Goals of this course

To improve your ability in the following areas:

1. Developing an organized methodology for implementing medium-large software systems
2. Team programming
3. Thinking innovately
4. Independent learning of new concepts, tools, and software frameworks
5. Oral and written communication and discussion of your ideas, designs and implementations

To master technical topics and tools of the software engineering trade:

1. The software development lifecycle: requirements; design; implement; test; deploy
2. Design and coding methodologies, including Object-Oriented Design, Design Patterns, Refactoring, and UML
3. Standard software engineering tools and frameworks, including Eclipse or IntelliJ, JUnit, JavaDoc, Maven, and git
4. Web programming basics including JSON, RESTful web servers and AJAX
5. Frameworks and tools you will learn on your own for your group project (varies by project topic)

Prerequisites are Data Structures (600.226) or significant Java programming experience. The course is generally not recommended for sophomores, but may be appropriate for the few sophomores that have significant programming experience.
git ≠ GitHub
WHY DO WE CARE
brownies

master
brownies

master
master

brownies
Object-Oriented Software Engineering

600.321/421 Prof. Scott Smith Monday/Wednesday 1:30-2:45, Shaffer 303

Goals of this course

To improve your ability in the following areas:

1. Developing an organized methodology for implementing medium-large software systems
2. Team programming
3. Thinking innovately
4. Independent learning of new concepts, tools, and software frameworks
5. Oral and written communication and discussion of your ideas, designs and implementations

To master technical topics and tools of the software engineering trade:

1. The software development lifecycle: requirements; design; implement; test; deploy
2. Design and coding methodologies, including Object-Oriented Design, Design Patterns, Refactoring, and UML
3. Standard software engineering tools and frameworks, including Eclipse or IntelliJ, JUnit, JavaDoc, Maven, and git
4. Web programming basics including JSON, RESTful web servers and AJAX
5. Frameworks and tools you will learn on your own for your group project (varies by project topic)

For more information on how the course will run -

Prerequisites are Data Structures (600.226) or significant Java programming programming experience. The course is generally not recommended for sophomores, but may be appropriate for the few sophomores that have significant programming experience.