Introduction to Programming Methods

Welcome to CS 2!

All About Me!

- My pronouns are **they/them**.
- Call me "Adam" or "Prof. Blank".
- I **care** about your experience in this course and at Caltech.
- CS 2 is my **favorite** course to teach!
- I love my dog **Hopper**.

Outline

- CS is for **EVERYONE**!!!!!!

Course Philosophy

- CS is for **EVERYONE**!!!!!!
- Aim is to be **practical** and **useful**

Course Philosophy

- CS is for **EVERYONE**!!!!!!
- Aim is to be **practical** and **useful**
- Centered around **labs** and **projects**

Course Philosophy

- CS is for **EVERYONE**!!!!!!
- Aim is to be **practical** and **useful**
- Centered around **labs** and **projects**
- Active Learning

- CS is for **EVERYONE**!!!!!!
- Aim is to be **practical** and **useful**
- Centered around **labs** and **projects**
- Active Learning
- Difficult, time-consuming, but well-supported

Course Philosophy

- CS is for **EVERYONE**!!!!!!
- Aim is to be **practical** and **useful**
- Centered around **labs** and **projects**
- Active Learning
- Difficult, time-consuming, but well-supported
- Programming language is an implementation detail

Just A Taste

Some of the labs/projects you will be implementing are:

- Mini Google Maps
- A scheme to hide text in images
- A guitar sound synthesizer
- An Othello bot

Lab

- Weekly "lab" like a chemistry or physics lab
- Practice that fits between lecture and projects
- Attendance is mandatory (lowest lab grade is dropped)
- You must attend the section you're signed up for due to space constraints

Course Structure

Lab

- Weekly "lab" like a chemistry or physics lab
- Practice that fits between lecture and projects
- Attendance is mandatory (lowest lab grade is dropped)
- You must attend the section you're signed up for due to space constraints

Projects

- These are what you'd usually call "sets"
- Some will be partner assignments
- Some will span multiple weeks
- This is where you will spend the bulk of your time in this course

Course Structure

Lab

- Weekly "lab" like a chemistry or physics lab
- Practice that fits between lecture and projects
- Attendance is mandatory (lowest lab grade is dropped)
- You must attend the section you're signed up for due to space constraints

Projects

- These are what you'd usually call "sets"
- Some will be partner assignments
- Some will span multiple weeks
- This is where you will spend the bulk of your time in this course

Office Hours

- Adam holds six office hours a week.
- Feel free to set up a private appointment if you prefer.
- TAs hold office hours from 7:30pm to midnight every day (including weekends).

During the course, we will...

- do lots of programming
- create a foundation for further study in CS
- create a foundation for using CS in other fields
- see lots of applications of CS to various fields
- have fun (???)

During the course, we will...

- do lots of programming
- create a foundation for further study in CS
- create a foundation for using CS in other fields
- see lots of applications of CS to various fields
- have fun (???)

After the course, you will be able to...

- program a medium-sized project on your own or in a group
- do CS interviews
- structure data to handle large data sets

During the course, we will...

- do lots of programming
- create a foundation for further study in CS
- create a foundation for using CS in other fields
- see lots of applications of CS to various fields
- have fun (???)

After the course, you will be able to...

- program a medium-sized project on your own or in a group
- do CS interviews
- structure data to handle large data sets

This is the course where you stop thinking like a "programmer" and start thinking like a Computer Scientist!

Course Website

`https://debuggi.ng`

Grading

To pass the course, you must:

- Get $\geq 70\%$ on project03, project05, and project07
- Average at least 80% or higher on the projects (excluding project08)
- Average at least 85% on the labs (with one lab dropped).

Lecture Materials

- All slides and code from lecture will be available after lecture has occurred.

Lecture Materials

- All slides and code from lecture will be available after lecture has occurred.
- This is an in-person class. There will be no recordings of lecture provided.

The Big Picture

- "Duck" types vs. explicit types

```
variable = "hello"
```

```
String variable = "hello";
```

The Big Picture

- "Duck" types vs. explicit types

```
variable = "hello"
```

```
String variable = "hello";
```

- White-space vs. braces

```
1 if condition:
2     do_something()
3 else:
4     do_something_else()
```

```
1 if (condition) {
2     do_something();
3 }
4 else {
5     do_something_else();
6 }
```

The Big Picture

- "Duck" types vs. explicit types

```
variable = "hello"
```

```
String variable = "hello";
```

- White-space vs. braces

```
1 if condition:
2     do_something()
3 else:
4     do_something_else()
```

```
1 if (condition) {
2     do_something();
3 }
4 else {
5     do_something_else();
6 }
```

- Procedural programming vs. object-oriented programming
  - Python: code does not need to be in a function
  - Java: not only does code need to be in a function, but all functions must be in a class