# Welcome to Computer Architecture!

# Introductions!

As a convention in this class, say your name before you speak to the group!

Name, Year, College

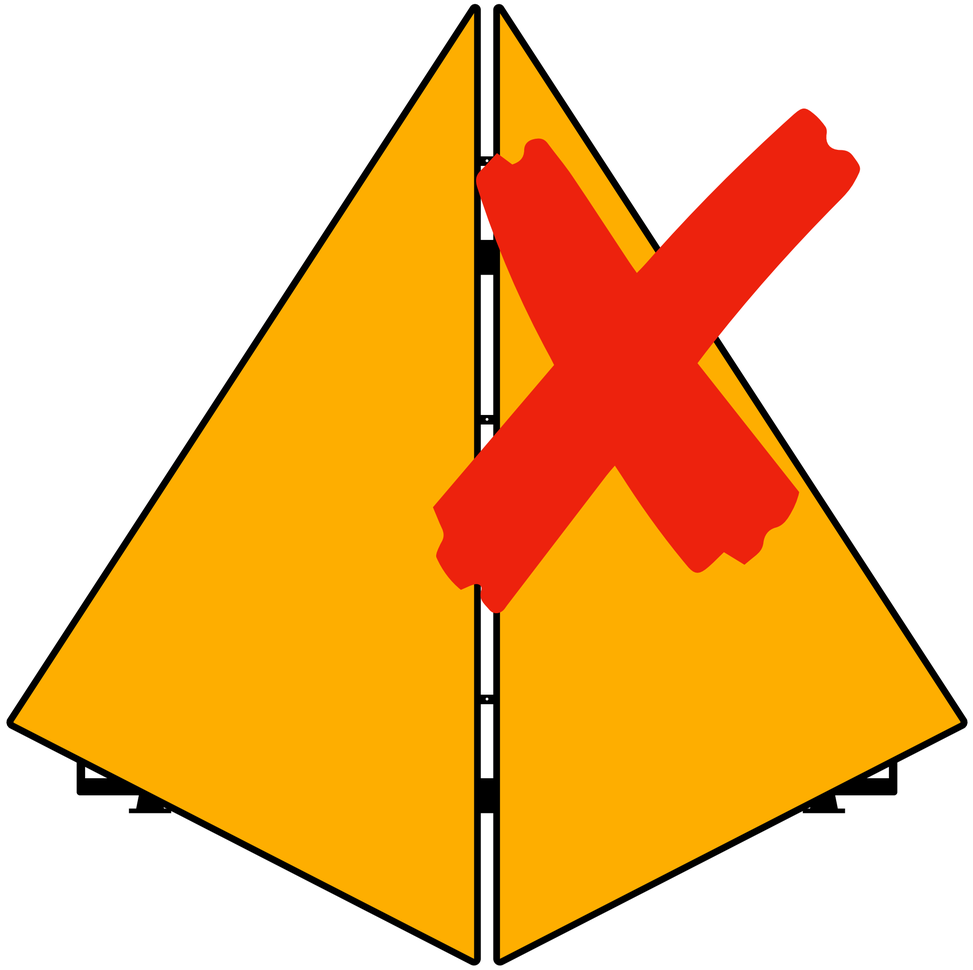
A recent class that you loved (doesn't have to be computer science!)

Something that you are looking forward to this semester

# Chat with your neighbor(s)!

So, how does a computer actually run a program? Be as detailed as possible!

# What is computer architecture?



😢

# What computer architecture *actually* is

Like almost every

“Computer archite

specifically, how s

Three general part

**Computer architecture gets fun when we consider the *interaction* between these parts!**

- ISA (instruction set architecture): what instructions can the computer execute and how are they defined?
- Microarchitecture: how does the CPU actually implement the ISA?
- Hardware system: what physically makes up the computer?

# Questions we'll answer

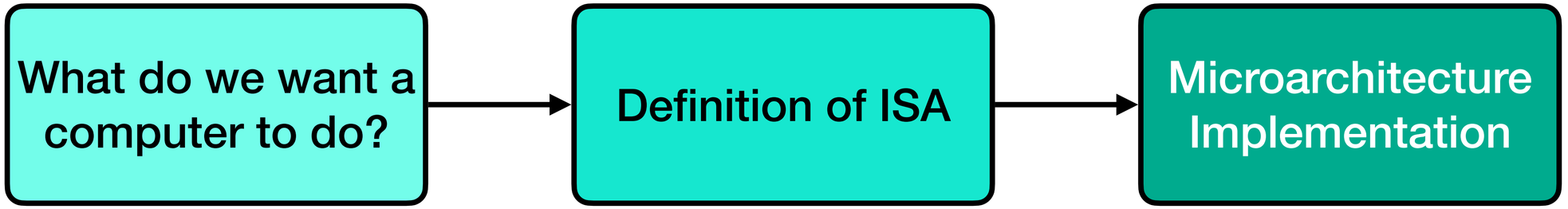
- How is a program represented on a computer?
- What is a CPU? How does it work?
  - stored bits to instructions?
  - instructions to tasks?
- How is data efficiently stored and used?
- What is the relationship between hardware and software?
- How does hardware use parallelism for efficiency?
- What's the history of the field? How do we feel that history today?
- Is my hardware secure? What should I be scared of? What could be done?
- What's the relationship between hardware and the environment?

# Chat with your neighbor(s)!

How do we measure whether one computer is “better” than another? Get creative — go beyond speed!

# How will we study these things?

- Try to “invent” them ourselves as much as possible!




- Simulate!
  - You will use C++ simulators and emulators: custom scaffolding and `gem5` for more complicated designs
  - Allows us to evaluate design choices!

**Caveat: we need to understand our simulators and their limitations to appreciate the takeaways!**

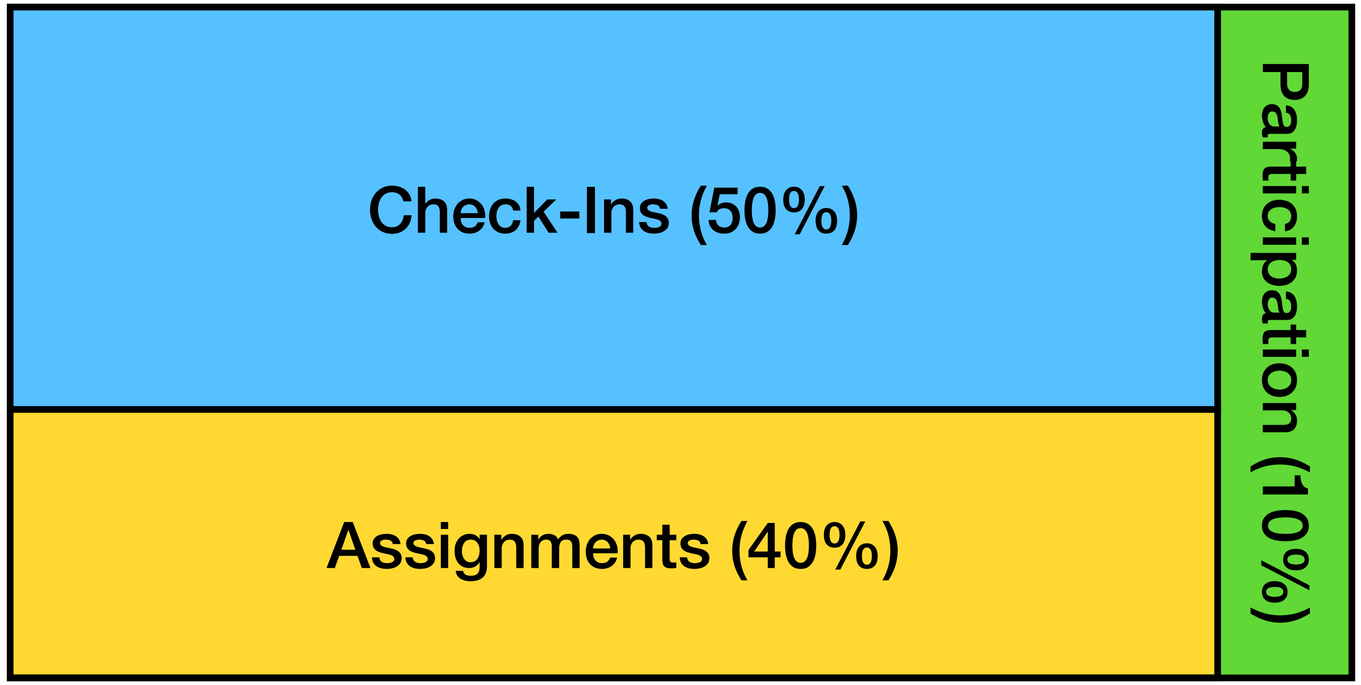
# What *won't* be taught in this course…

- We use C/C++ programming languages as a tool to study computer architecture. While you will have ample opportunity to practice, this is *not* a C/C++ class!
  - You should feel comfortable using malloc/new and free/delete, etc
  - You should feel comfortable using a debugger
- Converting binary to decimal, basic binary arithmetic, binary operators
  - We will cover these concepts very quickly so that we can get to the cool stuff!

# Course Structure and Policies

- Course website: https://cs.pomona.edu/classes/cs181ca/

# Grading



# Check-Ins (standards-based grading)

## Check-In 1.

Qu…r sit…ng elit. Adipiscing elit qu…ae

*In…is c…n sed diam …tempor.*

Qu…iring…um egestas. Iaculis ma…integ…

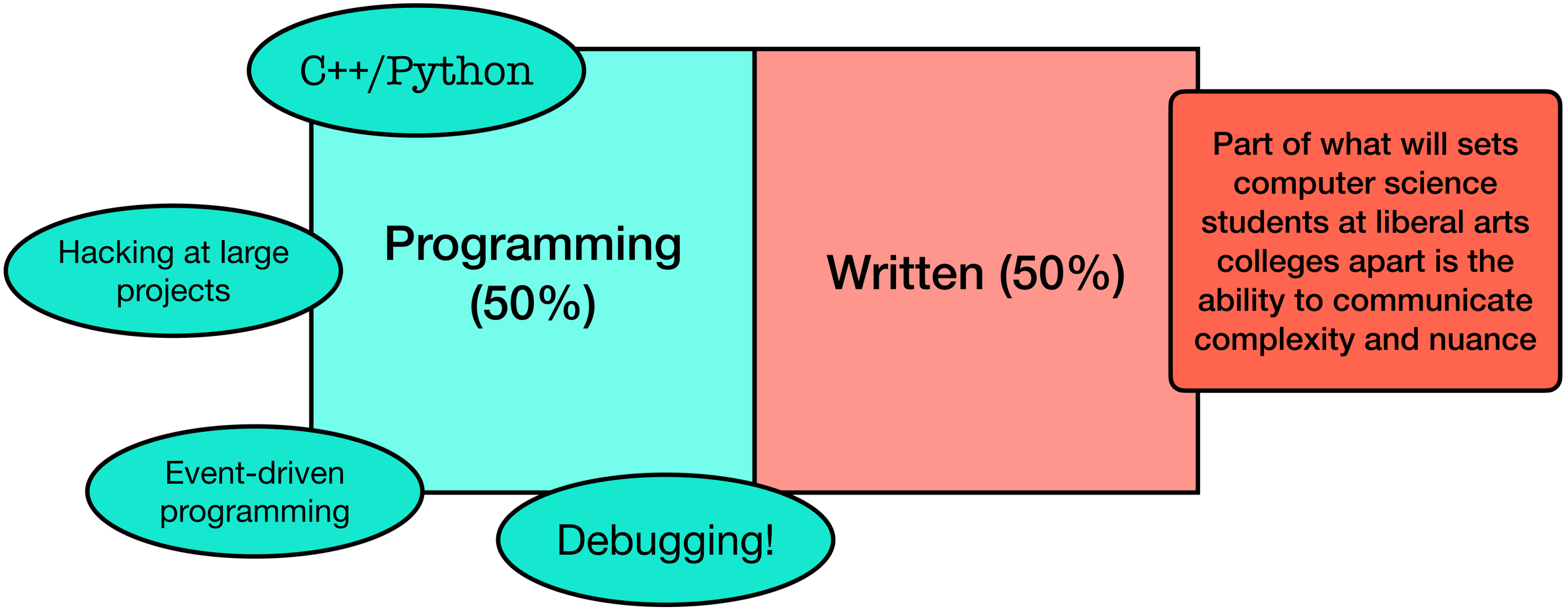
*Ut hendrerit semper vel class aptent taciti sociosqu. Ad lit…a torquent per conubia nostra inceptos himenaeos.*

Why do this?

What does this mean?

This isn't what I was looking for because…

# Assignments



# Feedback

- E-mail
- In-person (after class, in office hours)
- Anonymous feedback form on course website
- Department and college-wide resources

**Hold me accountable! Feedback only works if I follow up on it**

# Chat with your neighbor(s)!

Talk about some of the AI course policies you've used so far. What seems to work well? How has your personal usage aligned/failed to align with these policies?

# Let’s build the AI Policy for CS181CA!

# Group Discussion Policy, Things to Think About

- Constructing an inclusive environment in which discussion participants are encouraged to take risks;
- Ensuring discussion participants come to the discussion prepared to be effective participants;
- Initiating discussion with engaging, relevant topics or challenges;
- Encouraging active listening;
- Helping participants digest what they are hearing;
- Managing and facilitating the flow of the session;
- As needed, help the group reach a satisfactory closure point.

# Let’s build the Discussion Policy for CS181CA!