Life after CS106AP

CS106AP Lecture 28
Today’s questions

What have we accomplished so far in CS106AP?

What does my computer science journey look like after CS106AP?
Today’s topics

1. Honor Code reminder
2. Week 8 overview
3. Life after CS106AP
Honor Code
A reminder about the Honor Code

- You must cite all external resources you refer to (besides talking to course staff).
  - Talking to friends about high level ideas, looking at Wikipedia to better understand a concept, or searching for a Python error on StackOverflow – these are all natural parts of the collaborative nature of programming.
  - Giving credit is an extremely important part of real-world CS!
  - Especially on the Assignment 6 graphics extension – it’s totally fine to use online resources to help you implement your ideas. Just cite them!

- Takeaway: **Cite any help you receive!**
  - If you collaborated in an acceptable way and cited help, you haven’t violated the Honor Code!
A reminder about the Honor Code

- *Retroactive citations* due by **Friday, August 16 at 11:59pm**
  - If you forgot to cite someone/something, send us an email with the assignment and the resource you used/people you collaborated with.
  - Be as specific as possible!
- If you are in a more serious situation where you submitted an assignment that copied a friend’s code or took parts of an online solution, this is a violation of the Honor Code. **This will not apply to the vast majority of you**, but if you are in this boat, we will also allow **full assignment retractions**.
  - If you choose to retract an entire assignment, you will get an automatic 0 on that assignment with no questions asked.
  - Also due Friday, August 16 at 11:59pm.
In summary...

- Cite your sources on Assignment 6!

- Retroactive citations (and assignment retractions if necessary) must be submitted by this Friday at 11:59pm.

- If you have any questions about the Honor Code, please come to OH sometime this week (there are lots of opportunities this week!). We will not be answering any Honor Code questions over email.
Week 8 overview
Week 8

- Section this week is optional exam review
  - Section leaders may reschedule to times that work better for everyone in their sections
- Today: Life after CS106AP
- Tuesday: Final exam review
- Wednesday: In-class office hours
  - Attendance optional
  - Bring questions about the practice exam
- Thursday: Study day (no class)
  - Sonja will still have OHs, but they have been shifted to 10am-12pm
- Friday: Final exam
  - 8/16, 3:30pm to 6:30pm in Hewlett 200
  - No alternate exam times except for OAE accommodations
Final exam logistics

● Topics will cover those through Lecture 25 (up to and including last Tuesday)
● Topics that **will not** be on the exam
  ○ TKinter (only campy will be on the exam)
  ○ Jupyter
  ○ matplotlib
  ○ Doctests
  ○ How your computer works + Internet
  ○ Karel
  ○ GImages
● Two pages of notes front and back (4 sides)
● Practice exam released by Tuesday morning
What have we accomplished so far in CS106AP?
Our learning goals for the quarter

- I am excited to use programming to solve real-world problems I encounter outside class, including those related to my major/career.
- I can break down complex problems into smaller subproblems by applying the logical reasoning skills I have gained from programming.
- I better understand the technology in my everyday life and can identify the programmatic concepts present in these technologies.
- I understand how computers process and organize information.
Your accomplishments this quarter

You’ve learned...

- Control flow
- Functions and variables
- Python data types: ints, floats, strings, and more
- Data structures: lists, dictionaries, tuples
- Image processing
- File reading and parsing
- Console programs and command line arguments
- Graphics and event-driven programming
- Classes and object-oriented programming
- List and dictionary comprehensions
- Lambda functions and custom sorting
- Jupyter notebooks

You’ve built and implemented...

- An intelligent, problem-solving robot
- Educational software
- Image processing software
- Interesting and insightful data visualizations
- An arcade game
- An explorable explanation of real-world phenomena
Questions you can now answer

● What is possible with technology and code? What isn’t possible? How can I use programming to solve problems that I otherwise would not be able to?

● What makes “good” code and what are the best practices for writing code?
What does life look like after CS106AP?
What can I study within computer science?

(concentrations within CS)
Concentrations in computer science (at Stanford)

- Artificial Intelligence

Possible applications for Artificial Intelligence:

- Virtual Personal Assistant
- Natural language processing
- Deep Learning
- Simulation Modelling
- Machine Translation
- Visualization
- Audio Analytics
- Graph Analytics
- Image Analytics
- Internet of Things
- Robotics & Soft Robotics
- Machine Learning

source: statista via @mikequindazzi
Concentrations in computer science (at Stanford)

- Artificial Intelligence
- Biocomputation
Concentrations in computer science (at Stanford)

- Artificial Intelligence
- Biocomputation
- Computer Engineering
Concentrations in computer science (at Stanford)

- Artificial Intelligence
- Biocomputation
- Computer Engineering
- Graphics
Concentrations in computer science (at Stanford)

- Artificial Intelligence
- Biocomputation
- Computer Engineering
- Graphics
- Human-Computer Interaction
Concentrations in computer science (at Stanford)

- Artificial Intelligence
- Biocomputation
- Computer Engineering
- Graphics
- Human-Computer Interaction
- Information
Concentrations in computer science (at Stanford)

- Artificial Intelligence
- Biocomputation
- Computer Engineering
- Graphics
- Human-Computer Interaction
- Information
- Systems
Concentrations in computer science (at Stanford)

- Artificial Intelligence
- Biocomputation
- Computer Engineering
- Graphics
- Human-Computer Interaction
- Information
- Systems
- Theory
What can I do with computer science?

(applications of CS)
Applications in CS 106AP

- Khansole Academy
  - Scalable education + biocomputation
Applications in CS 106AP

- Khansole Academy
  - Scalable education + biocomputation
- Fun with Images
  - Computer-generated art, image processing in environmental context
Applications in CS 106AP

- Khansole Academy
  - Scalable education + biocomputation

- Fun with Images
  - Computer-generated art, image processing in environmental context

- BabyNames
  - Social data visualization
Applications in CS 106AP

- **Khansole Academy**
  - Scalable education + biocomputation

- **Fun with Images**
  - Computer-generated art, image processing in environmental context

- **BabyNames**
  - Social data visualization

- **Breakout**
  - Video games
UNESCO Sustainable Development Goals
UNESCO Sustainable Development Goals

- Zero hunger
- Good health and well-being
- Quality education
- Clean water and sanitation
- Affordable and clean energy
- Industry, innovation, and infrastructure
- Sustainable cities and communities
- Climate action
- Peace, justice, and strong institutions
UNESCO Sustainable Development Goals

- Zero hunger
  - Monitoring and predicting crop yield
  - Predicting and preventing crop diseases
UNESCO Sustainable Development Goals

● Good health and well-being
  ○ Biomedical computation
  ○ Pharmacogenetics
    ■ how an individual will respond to a drug
  ○ Mental health
    ■ CS isn’t universally good
      ● *it’s about what you do with it*
UNESCO Sustainable Development Goals

- Quality education
  - Scalable education, adaptive learning (personalized)
  - Access to a computer == access to a world of education
UNESCO Sustainable Development Goals

- Clean water and sanitation
  - Modeling flow of water through nanotubes
    - for more efficient filters
UNESCO Sustainable Development Goals

- Affordable and clean energy
  - Maximizing wind energy
  - Logistics of moving wind turbines
UNESCO Sustainable Development Goals

- Industry, innovation, and infrastructure
  - Infrastructure: self-driving cars
UNESCO Sustainable Development Goals

- Sustainable cities and communities
  - Computational sustainability
    - Smart grid
    - Intelligent transportation systems
UNESCO Sustainable Development Goals

- Climate action
  - Modeling climate and extreme weather events
UNESCO Sustainable Development Goals

- Peace, justice, and strong institutions
  - Security
UNESCO Sustainable Development Goals

- Zero hunger
- Good health and well-being
- Quality education
- Clean water and sanitation
- Affordable and clean energy
- Industry, innovation, and infrastructure
- Sustainable cities and communities
- Climate action
- Peace, justice, and strong institutions
How does Python translate to other programming languages?
Other programming languages: Python

evens = []
for i in range(100):
    if i % 2 == 0:
        evens.append(i)
print(evens)
Other programming languages: Python

evens = []
for i in range(100):
    if i % 2 == 0:
        evens.append(i)
print(evens)

# With a list comprehension instead
print([i for i in range(100) if i % 2 == 0])
Other programming languages: JavaScript

```javascript
var evens = [];
for(var i = 0; i < 100; i++) {
    if(i % 2 === 0) {
        evens.push(i);
    }
}
console.log(evens);
```
Other programming languages: Java

```java
ArrayList<Integer> evens = new ArrayList<Integer>();
for(int i = 0; i < 100; i++) {
    if(i % 2 == 0) {
        evens.add(i);
    }
}
System.out.println(evens);
```
Other programming languages: C++

```cpp
using namespace std;
vector<int> evens;
for(int i = 0; i < 100; i++) {
    if(i % 2 == 0) {
        evens.push_back(i);
    }
}
cout << evens << endl;
```
Takeaways

● Computer languages are all very similar and include many of the same structural features!
● What’s different is their **syntax**.
  ○ Python is “light” in syntax.
  ○ Other languages just have more to type in – heavier syntax allows for more auto-error checking in languages like C++.
● Once you learn one language, others are easier to pick up.
  ○ They may look different or seem more complex, but soon you’ll recognize the familiar structures.
Programming paradigms

- Imperative vs. declarative

*Focus on how things happen*
Programming paradigms

- Imperative vs. declarative

- Focus on how things happen
- Focus on what happens
Programming paradigms

- Imperative vs. declarative
  - All of the previous examples are **imperative** programming languages.
    - Object-oriented programming (classes, objects, etc.) is one of many imperative programming paradigms.
  - HTML and CSS are **declarative** programming languages
    - Put the text on the screen and say what each item is/what it will look like.
Programming paradigms

- Imperative vs. declarative
  - All of the previous examples are imperative programming languages.
    - Object-oriented programming (classes, objects, etc.) is one of many imperative programming paradigms.
  - HTML and CSS are declarative programming languages
    - Put the text on the screen and say what each item is/what it will look like.

**Takeaway:** You may encounter different programming paradigms, depending on what language you’re using!
What did you ask us...
What are the best courses in CS for getting a job after my Stanford education?
What are the best courses in CS for getting a job after my Stanford education?

- CS 106B - Programming Abstractions
  - Many interview problems are CS106B section problems.
  - The concepts you learn and technical maturity you develop in CS106B equip you for the demands of many industry jobs.

- CS 161 - Design and Analysis of Algorithms
  - Gives you a deeper understanding of the algorithms you implement in CS106B.
  - Also often-asked during interviews.
Which Stanford courses did you enjoy a lot? Are there any that changed your lives?
Which Stanford courses did you enjoy a lot? Are there any that changed your lives?

- Educ 236, Educ 208B, CS 206, CS 124, English 12A, English 157
- History 50B, Africaam 47
- Music 24a/b/c, Music 122B, Psych 70, CS 109, CS 231N
What is a good way to develop CS skills and learn to apply them in my work as a grad student? I feel like I started too late to be able to catch up.
What is a good way to develop CS skills and learn to apply them in my work as a grad student? I feel like I started too late to be able to catch up.

- It’s never too late to learn CS.
- Take more courses!
- Spend time coding/developing your own projects
  - Use online resources to learn (e.g. tutorials, courses, YouTube channels)
- Seek out mentorship/collaborators and get hands-on experience
How is CS used professionally?
How is CS used professionally?

- Data analyst
- Software engineer
- Research scientist
- Project manager
- UI/UX designer
- ... and so much more!
What other programming languages are key to learn other than Python?
What other programming languages are key to learn other than Python?

- A few we’ve already talked about...
  - C++ and Java (object-oriented programming)
  - JavaScript (web programming)
- A few others...
  - HTML and CSS (more web programming)
  - Swift and Android (mobile programming)
How can we continue learn more coding without taking more courses? How can we improve our coding skills on our own?
How can we continue learn more coding without taking more courses? How can we improve our coding skills on our own?

- Some online resources (mostly in the form of free courses)
  - Codecademy
  - Coursera, edX, Udemy, and other MOOCs
  - Khan Academy
  - MIT Open Courseware

- Strategies for programming self-improvement
  - Write lots of code!
  - Work on a project you find inspiring
  - Find other people to collaborate with
    - Join an open-source project!
How do we export our code out to the greater Python community?
How do we export our code out to the greater Python community?

- Freezing your code
  - Bundle together your code, any dependent libraries, and the Python interpreter
  - Distribute a single, executable file to users

- Packaging your code
  - Bundle your code (libraries/modules) and make it available online
  - Open source packages (like campy) are made available through PyPI
  - Other programmers can install your libraries using pip
What questions do you have?
What’s next?