CS 224G Introduction

Programming AI Apps in the LLM Age

John Whaley, Jan Jannink CS 224G - 1/6/2026

Administrative Details

- Instructors
  - Jan Jannink yan@stanford.edu, John Whaley jwhaley@stanford.edu
  - CA : Andrea Mock amock@stanford.edu
- If you haven't already gotten in, please request access to the slack
  - https://cs224g-winter-2026.slack.com/
- Office hours planned for (Wednesday 2-4PM)
  - Reach out to me for Zoom any time
- Feedback
  - How we adapt the class to your needs

Who we are, Why we are doing this

- Jan and John, CS PhDs, Serial Entrepreneurs, Instructors
- Exits in previous AI startups, UnifyID + Redcoat (John), imeem +VoiceBase (Jan)
- Founders of new AI startups, Synthpop (Jan), Inception Studio (John)

- LLM advances permit a re-envisioning of all tech startups
- Unique opportunity to share knowledge as the trend emerges

Review : CS224G Historical Timeline

- Nov 2022 : Chat GPT, Inception Studio cohort 1, Synthpop founders reconnect
- Feb 2023 : Llama initial launch (open weights)
- Mar 2023 : GPT-4 & Claude launch, course concept developed
- Jan 2024 : CS224G first edition, Mistral launch
  - Benchmark saturation
- Feb 2024 : GitHub Copilot, Gemini launch
  - Large contexts, multimodal
- Late 2024 : Bolt, Lovable
- Jan 2025 : CS224G v2, Visit from Grove AI, DeepSeek R1 launch
  - Reasoning model
- May 2025 : Claude Code
- Aug 2025 : Nano Banana
  - Video pipeline
- Jan 2026 : CS224G v3

Course Format

- 10 week crash course (bootcamp style)
- Dream up a project (this week)
- Form/Join a team (next week)
- Develop a solution
- 2 week sprints
- Demo every sprint for 4 sprints
- Finish with Demo Day Event

Course Style

- Agile, just in time approach
- Don’t expect everything to be fully prepared in advance
- Course Slack is the best way to stay up to date
- Presence and participation is a must!
    - We’re giving it our all, and so are the CAs
- The space is changing so quickly
- Other approaches would not do the topic justice

Course Timeline

- Jan. 8 Initial project proposal presentations
- Jan. 13 Project proposals due
- Jan. 15 All project teams formed (before start of first sprint)

- Jan. 29, Feb. 12, Feb 26 Project checkpoints (demos and push to GitHub)
- Mar. 11 Final Sprint Due
- Finals week: Demo Day and Poster Session (investor & entrepreneur review)

Project Proposals (due Jan. 13)

- Name
- Email
- GitHub alias (to submit projects)
- Project Topic/Subject
- Team members/open slots (2-4 recommended)

Example Proposal

Project Areas

- Multimodal apps: sound, language, image, video
- Data Extraction, Scraping, Form Filling
- Assistants, Interns, Copilots
- Personalization, Content Rewriting
- Coding, APIs
- Law, Medicine, Government (anything with manual document processing)

Project Grading

- 10% participation and presence (class, Slack)
- 45% spread over 3 project checkpoints
  - good instructions for testing
  - actual code contribution (git commits)
- 45% final project evaluation and presentation

- We'll evaluate feedback from the class, student and team contributions

Agile Project Methodology

- Quickly come up with something
- Present on Thursday
- Talk it up with the class on Slack
- Find similar proposals and interests
- Refine, revise based on conversations
- Form compatible teams, we don’t have room for solo projects
- More iterations almost always produce better outcomes

Picking a Good Project/Team

John Whaley, Jan Jannink CS 224G - 1/6/2026

What makes a great project?

- Impact
  - Solving a real problem for real people
  - Where LLMs can enable a meaningful solution
- Realistic scope
  - Bounded risk
  - Right-sized to a 10-week course
  - Meaningful intermediate milestones
- Team fit
  - What will keep you excited about working on this?
  - Something you already know + something you want to learn

What makes a great team?

- Team size of 2-4
  - 2 is possible, but tough
  - 3-4 is optimal
  - 5 means at least one person won't pull their weight
- Complementary skill sets
  - Backend, frontend, design, product management, communication
  - Make sure your team has strong coverage in ALL of these
- Alignment
  - Interest areas
  - Philosophy
  - Commitment levels and work expectations

First Project Ideas

John Whaley, Jan Jannink CS 224G - 1/6/2026

Project Matching

Add yourself and ideas to our project matching sheet

CS 224G Themes

John Whaley, Jan Jannink

CS 224G - 1/6/2026

Programmable AI

- Content mashups are easier than ever (nano banana + Veo, Suno)
- Streamlining of tedious repetitive manual work (Data Entry, coding)
- Tasks not amenable to traditional CS techniques

- There are also pitfalls to navigate (safety, stability, hallucination)
- We will learn much more exploring together

Chat Emergent Characteristics

- The chat paradigm is now:
- (de)composable
- programmable

- scalable
- complex : computationally complete

Nondeterministic Programming Model

- LISP introduced the deterministic functional programming paradigm
- ChatGPT heralds the arrival of nondeterministic functional programming

- Instead of bottom up software development, top down solution mining
- Instead of largely reliable CPU/GPU performance, finicky & arbitrary behavior

LLM as a Smart Overeager Inexperienced Intern

- Tremendous amount of knowledge
- Compelled to answer no matter what
- No experience combining facts
- Extremely literal
- Can be lazy, looks for shortcuts
- Mimics the style of the prompter

This Course as Managerial Training

How do you as an individual scale?

Congratulations you just acquired a team

We'll help you get it into shape

Build an innovative project at the same time

Learn what it takes to go beyond academia