CS1-LLM: Integrating LLMs into CS1 Instruction

Written by Annapurna Vadaparty, Daniel Zingaro, David H. Smith IV, Mounika Padala, Christine Alvarado, Jamie Gorson Benario, Leo Porter

Presented by Weixuan Liao & Arshiya Mahmoodinezhad

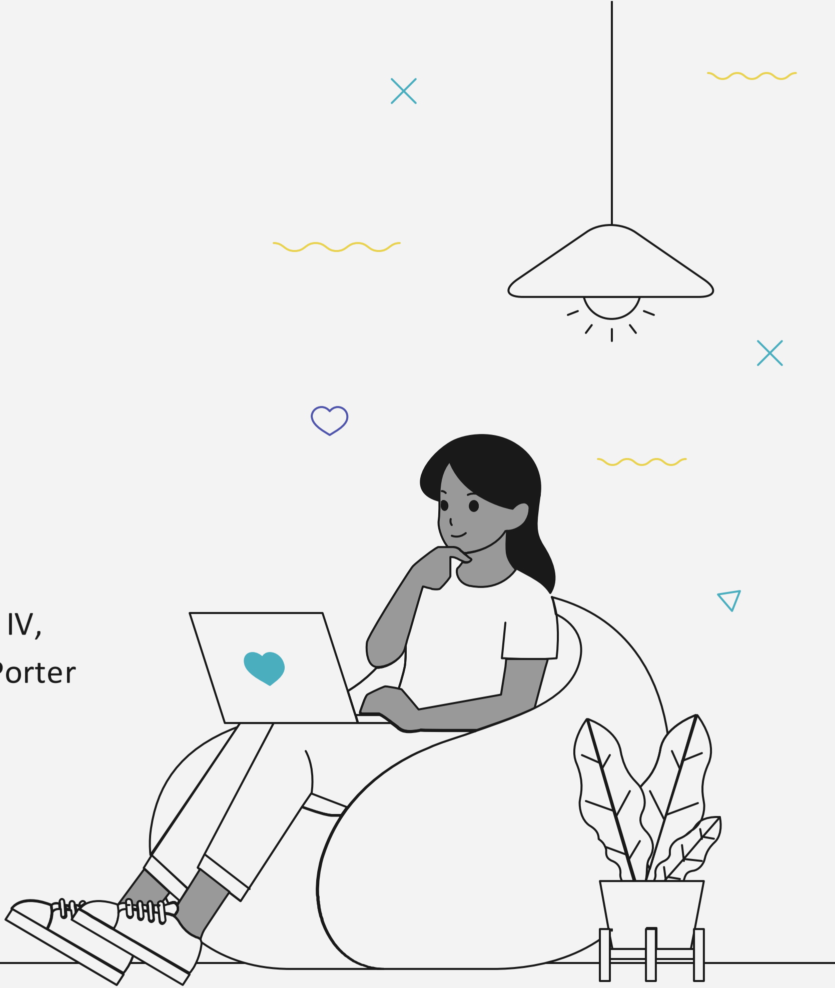


Table of contents

01 Introduction & Design

The motivation behind integrating LLMs into CS1 and the course redesign principles

02 Course Context

The course structure, key learning goals, scheduling and how LLMs were integrated to achieve set goals

03 Student Perception

Summary of student feedback on using LLMs, highlighting both positive and negative experiences

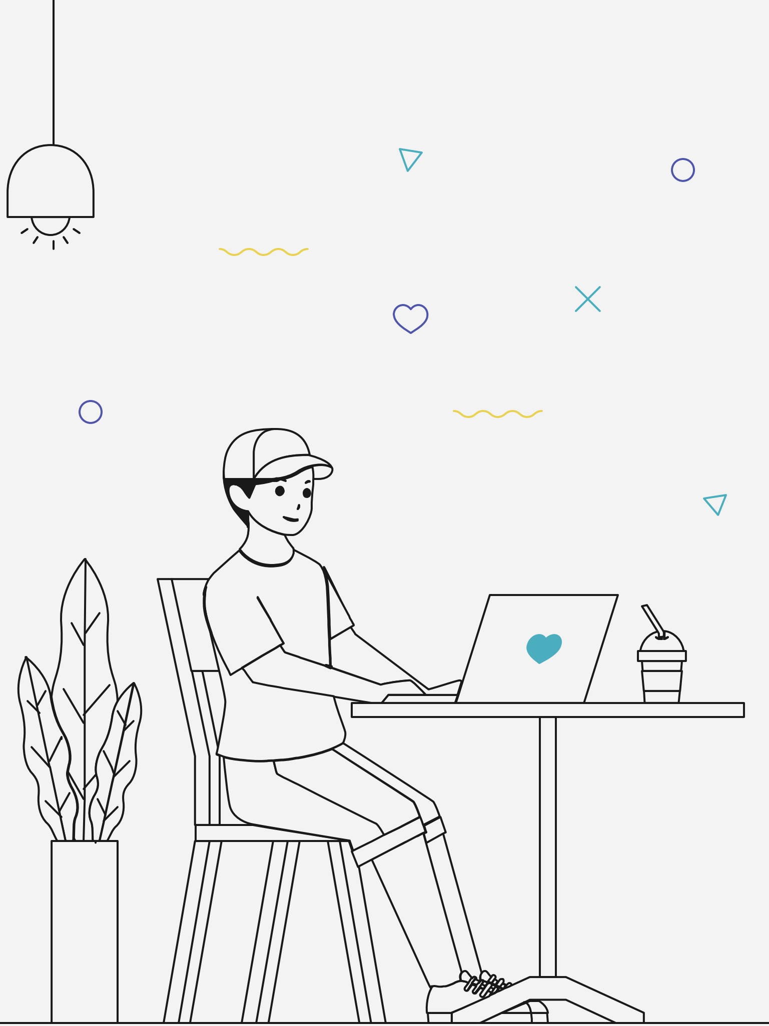
04 Discussion & Conclusions

Discussing key lessons learned from the course and provide recommendations for future iterations

01

Introduction

The motivation for integrating LLMs into the CS1 course, the changing landscape of AI, and how tools like GitHub Copilot are transforming programming education



The Future of Computing Education!

In 2023, research highlighted the growing impact of Large Language Models (LLMs) and Generative AI (GenAI) on computing education. Tools like GPT-4 can now solve programming problems at the level of top students, prompting educators to rethink course designs. Some are restricting AI use, while others are embracing the changes.

Redesigning Introductory Programming Courses!

They proposed reshaping introductory programming courses around Large Language Models (LLMs). With 92% of U.S. developers using LLMs, the skills needed for software development are evolving.

Students should learn with LLMs to prepare for a workforce that increasingly relies on these tools, shifting focus from writing code from scratch.

LLMs enable students to tackle larger, open-ended projects, improving engagement and real-world programming skills. Our new CS1 course uses LLMs to help students overcome syntax issues and focus on creative programming

Design Goals:

The CS1-LLM course was designed with five key principles to help students benefit from Large Language Models (LLMs) like GitHub Copilot.

1. **LLM Integration:** Students are encouraged to use LLMs throughout their coursework, including some supervised quizzes and tests. The course balances teaching coding fluency both with and without LLM assistance.

1. **Industry Preparation:** The course bridges the gap between academic coding practices and real-world industry needs, focusing on code reading, modification, and feature development rather than just writing standalone programs.

3. **Supporting Underrepresented Groups:** Best practices, like Peer Instruction and pair programming, are employed to improve outcomes for underrepresented students.

4. **Creativity in Assignments:** The course promotes open-ended projects to foster creativity, moving away from constrained problems typically used in auto-tested assignments.

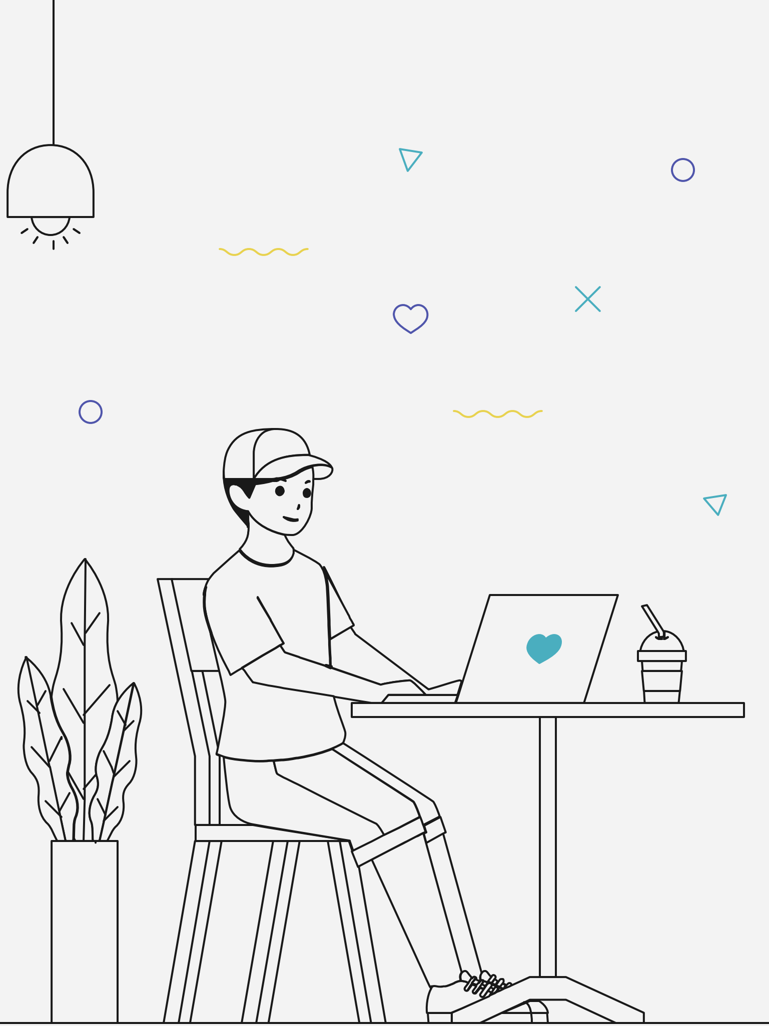
5. **Serving All Students:** The course serves both future CS professionals and those who won't pursue more CS courses, emphasizing high-level skills and practical applications over low-level syntax

02

Course CS1-LLM

Course context, learning goals, structure, scheduling and how LLMs were integrated to achieve set goals

Course Materials Link



Course Context

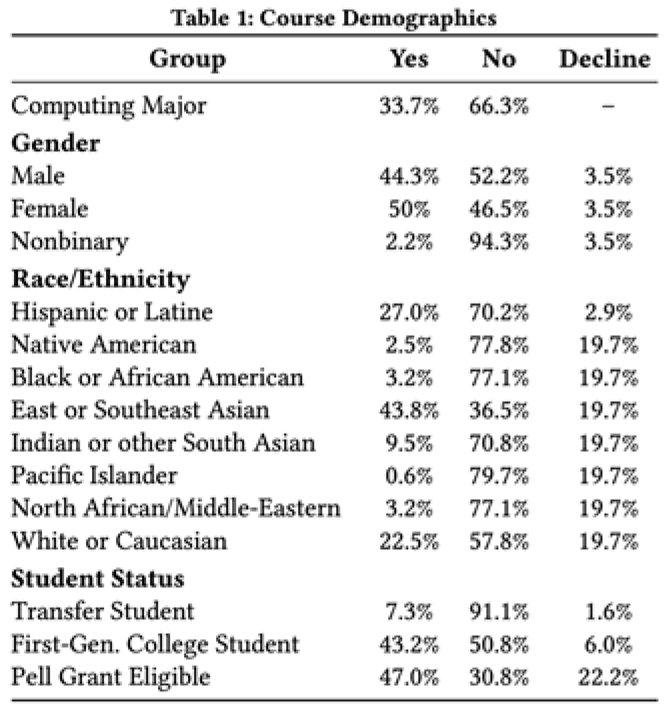


Table 1: Course Demographics

Group	Yes	No	Decline
Computing Major	33.7%	66.3%	–
Gender			
Male	44.3%	52.2%	3.5%
Female	50%	46.5%	3.5%
Nonbinary	2.2%	94.3%	3.5%
Race/Ethnicity			
Hispanic or Latine	27.0%	70.2%	2.9%
Native American	2.5%	77.8%	19.7%
Black or African American	3.2%	77.1%	19.7%
East or Southeast Asian	43.8%	36.5%	19.7%
Indian or other South Asian	9.5%	70.8%	19.7%
Pacific Islander	0.6%	79.7%	19.7%
North African/Middle-Eastern	3.2%	77.1%	19.7%
White or Caucasian	22.5%	57.8%	19.7%
Student Status			
Transfer Student	7.3%	91.1%	1.6%
First-Gen. College Student	43.2%	50.8%	6.0%
Pell Grant Eligible	47.0%	30.8%	22.2%

Classes include

- 3 hours of lecture
- 1 hour of in-person discussions
- 1 hour of closed labs

Diverse student population: including Computer Science, Computer Engineering, Bioinformatics, and Data Science. The demographics also reveal that 47% of students were Pell Grant eligible, indicating many from low-income backgrounds.

This course covers fundamental programming concepts like variables, functions, conditionals, loops, strings, lists and dictionaries in Python.

Course Learning Goals

01

Knowledge

Define nondeterminism, LLM, prompt, and related terms.

02

Comprehension

Illustrate AI workflow

Describe key Python features

03

Application

Use prompt engineering to guide AI code output.

04

Analysis

Analyze, divide, and debug Python programs effectively.

05

Synthesis

Design, test, fix, modify, and write Python programs.

06

Evaluation

Use testing evidence to verify program correctness.

Course Structure

Week 1-4:

- How to read, trace, and explain code
- How to use GitHub Copilot
- Basics (variables, conditionals, loops, functions, strings, and lists)

Week 5-10:

Transitioned to a more software engineering-focused approach, teaching skills like problem decomposition, testing, and debugging across three different domains: data science, image manipulation, and game development.

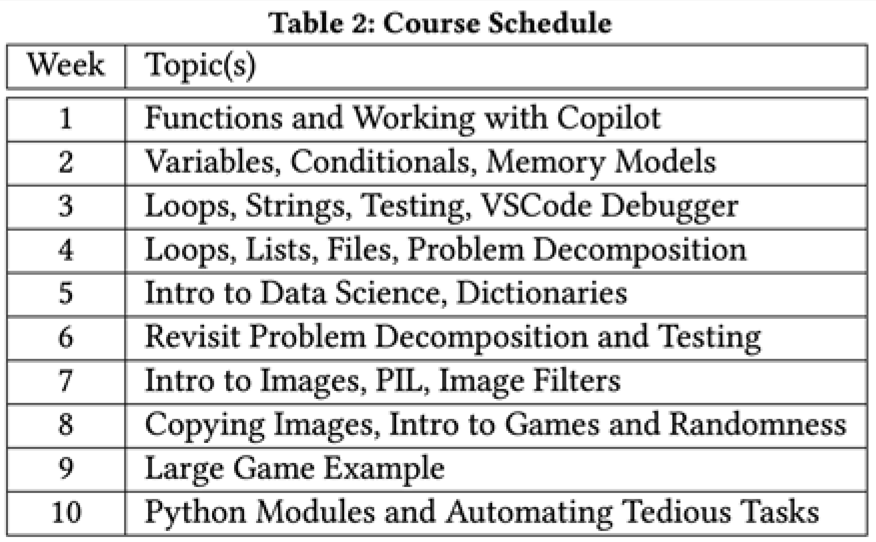
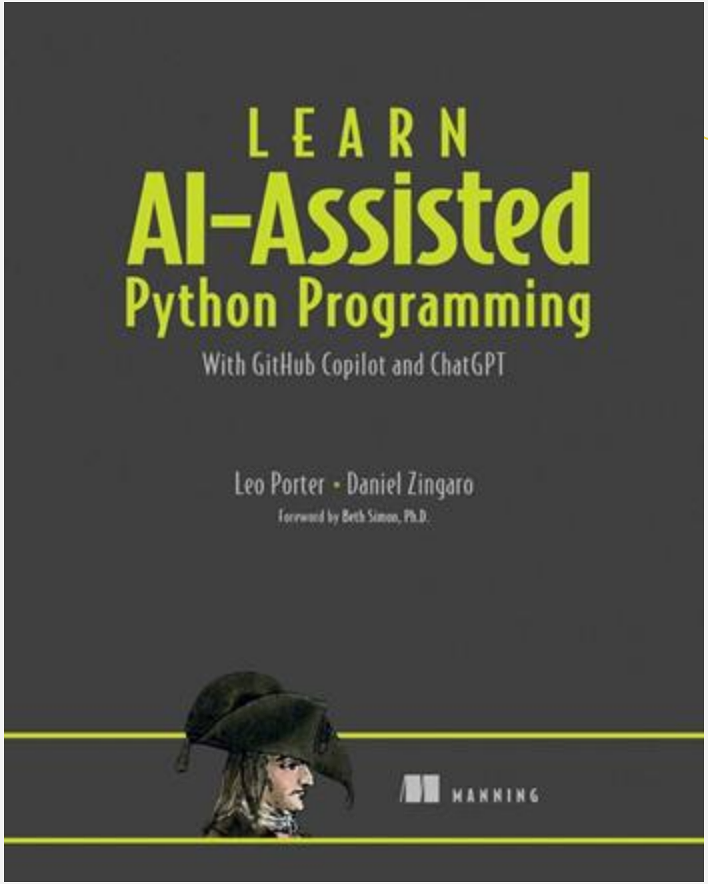


Table 2: Course Schedule

Week	Topic(s)
1	Functions and Working with Copilot
2	Variables, Conditionals, Memory Models
3	Loops, Strings, Testing, VSCode Debugger
4	Loops, Lists, Files, Problem Decomposition
5	Intro to Data Science, Dictionaries
6	Revisit Problem Decomposition and Testing
7	Intro to Images, PIL, Image Filters
8	Copying Images, Intro to Games and Randomness
9	Large Game Example
10	Python Modules and Automating Tedious Tasks

Textbook

Learn AI-Assisted Python Programming with GitHub Copilot and ChatGPT


LEARN
AI-Assisted
Python Programming
With GitHub Copilot and ChatGPT
Leo Porter • Daniel Zingaro
Foreword by Beth Simon, Ph.D.
MANNING

Nov, 2023

By Leo Porter, Daniel Zingaro

Lectures

Combination

- ❑ Mini-lectures
- ❑ Live Coding
- ❑ Peer Instruction

Changes

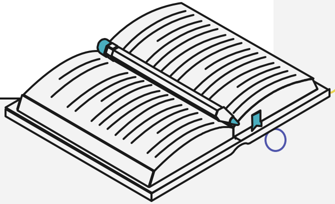
- ❑ Discussions on Copilot interactions
- ❑ Examples of incorrect Copilot responses
- ❑ Conversations with Copilot Chat

Assessments

Formative assessments accounted for 35% of the grade;

Summative assessments comprised the remaining 65%

Homework	**15%** Weekly, Multiple question types, No restrictions on using LLMs
Quizzes	**30%** Total four, 50-minute, Mostly without access to Copilot
Projects	**10%** One project per domain of data science, image manipulation, and games
Labs	**10%** Weekly, Synchronous/at home, some work with Copilot and some not
Final Exam	**25-30%** 90mins, 70%, Multi-choice component 45mins, 15%, Code writing tasks 45mins, 15%, One large new problem to be completed with Copilot

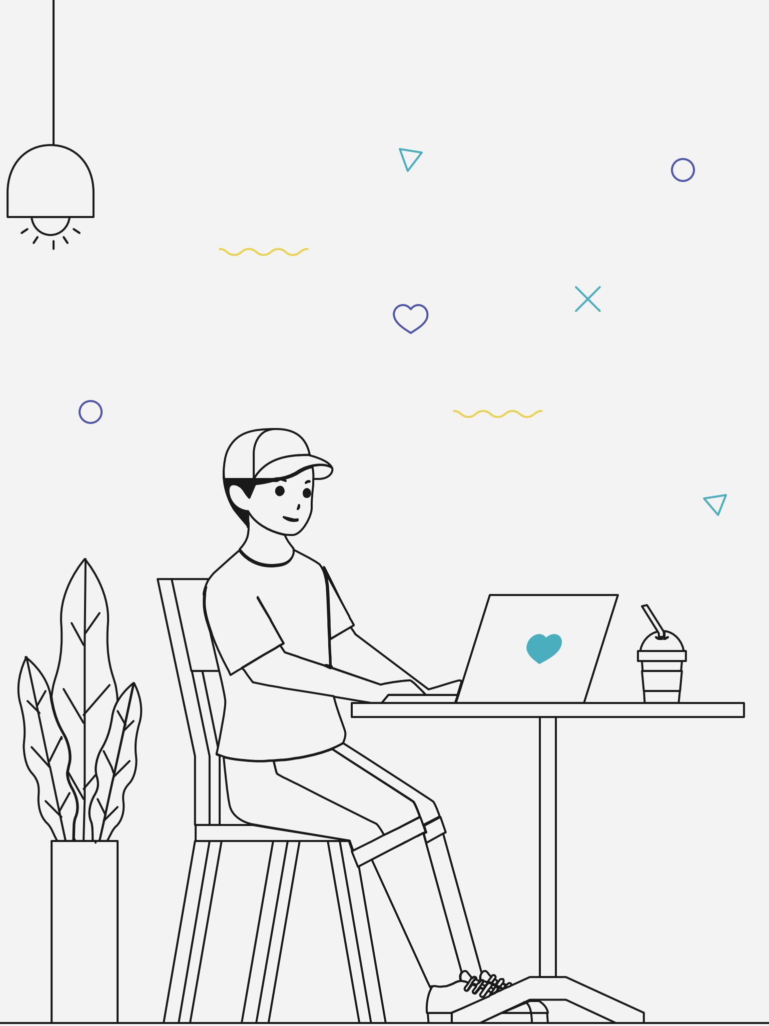


Readings and other online quizzes/surveys: 5% and Participation: 0-5%

03

Student Perceptions

To evaluate the impact of the redesigned CS1 course, instructors surveyed students, focusing on their experiences with LLMs



Student Comfort with Copilot

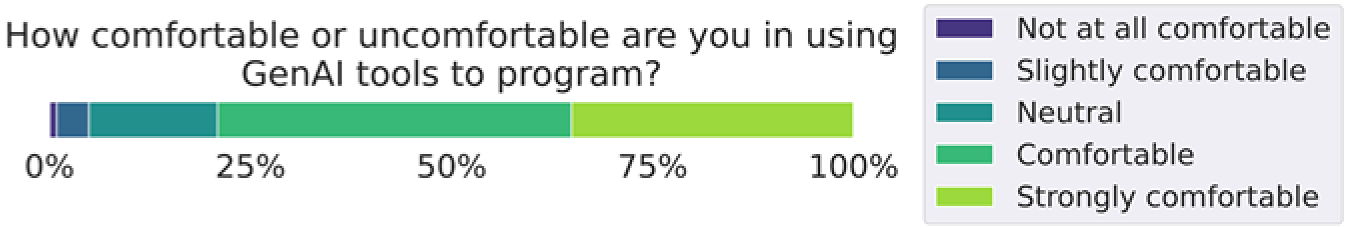


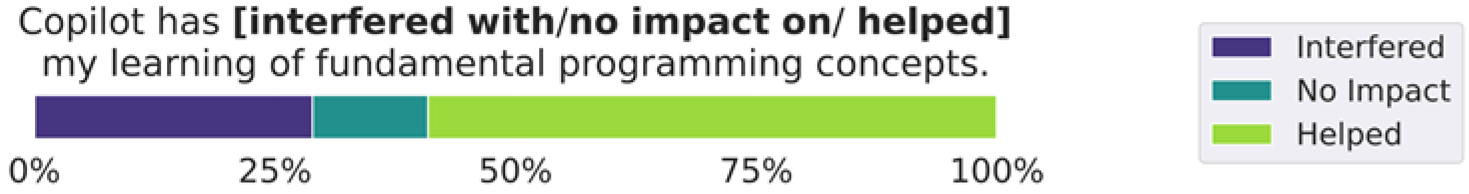
Figure 2: Student Comfort Programming with GenAI

79%

Comfortable

The majority of students felt comfortable using GenAI tools like Copilot.

Impact of Copilot on Student Learning



Copilot has **[interfered with/no impact on/ helped]** my learning of fundamental programming concepts.

Figure 3: Student Perceptions of how Copilot impacted their learning.

59%

Helped

A slight majority of students reported that it had helped them understand programming concepts.

“It was really nice having an assistant that could always help me the moment I needed it and made programming a lot less daunting.”

“If I were asked to code without Copilot, I wouldn’t feel very confident in myself despite doing well in the course.”

Student Confidence

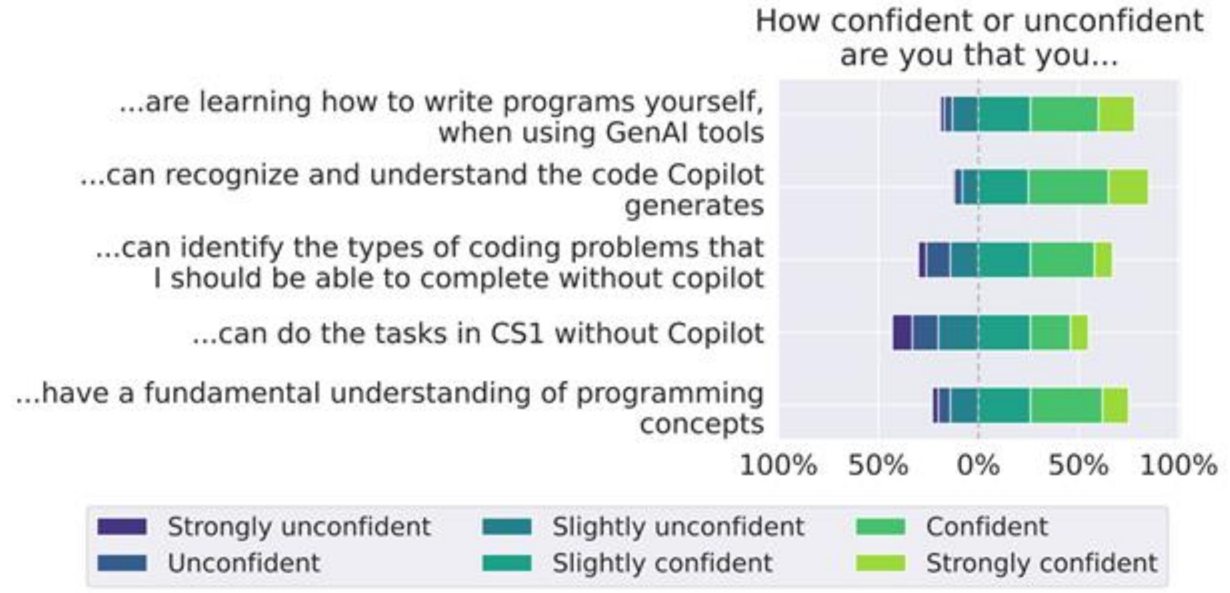


Figure 4: Student confidence in their ability and understanding at the end of the course.

31.1%

Low Confidence

Some students expressed low confidence in identifying coding problems they should be able to solve independently.

Some students found the inconsistency in using Copilot frustrating, as it was allowed in certain parts of the course but not in others, such as quizzes

How Students Interacted with Copilot

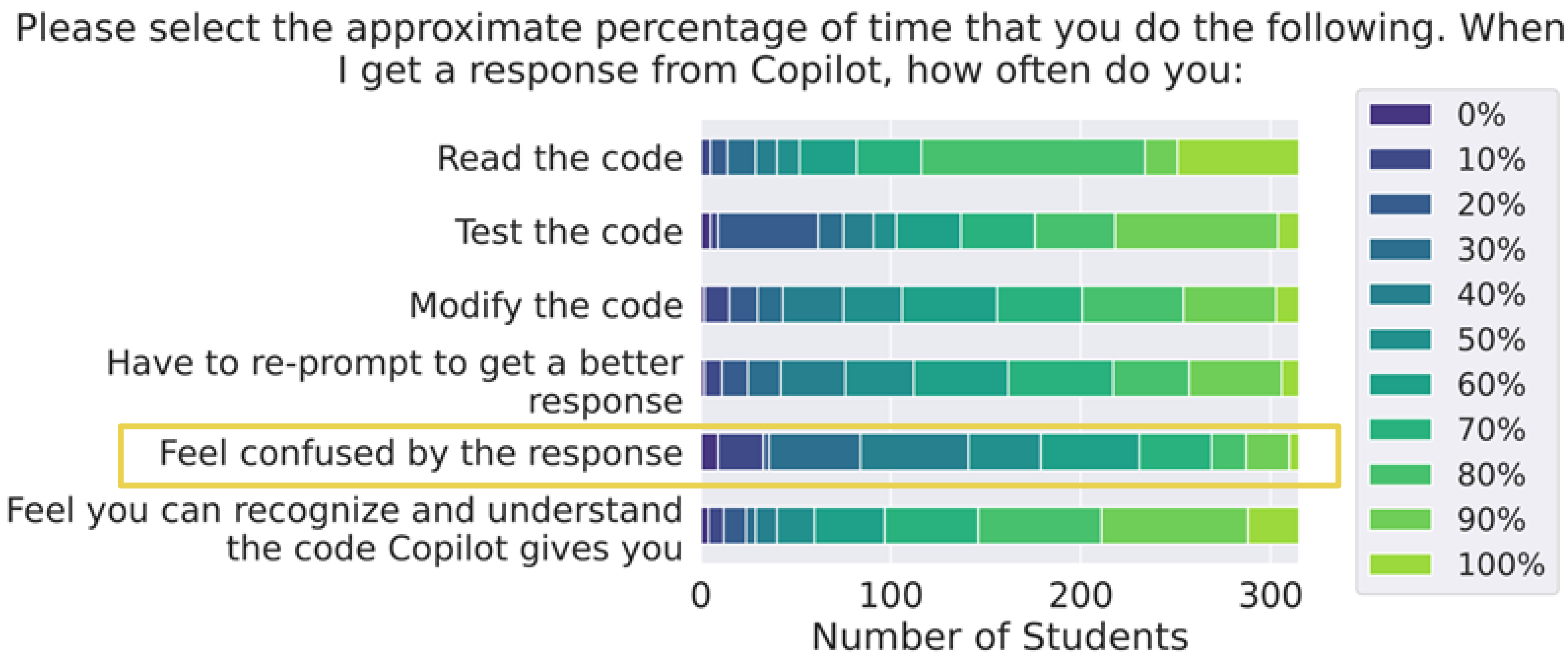


Figure 5: Student perceptions on how they interact with Copilot.

Student Comfort with Copilot

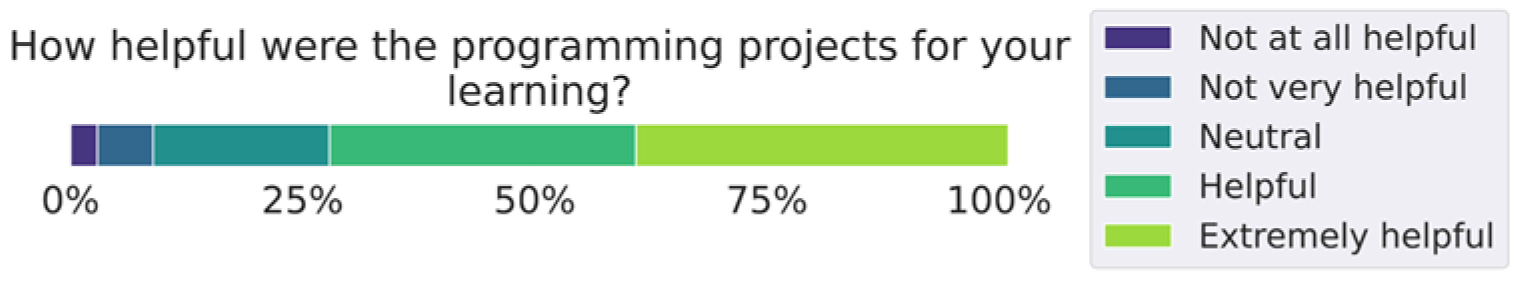


Figure 6: Students' View on Learning with Projects

70%

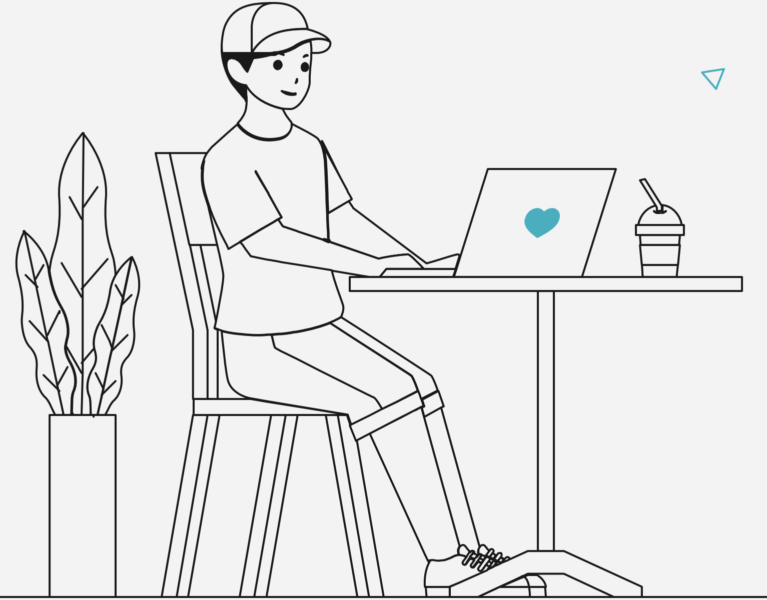
Helpful

Students generally found these projects valuable for their learning

"Though they were a little frustrating at times, I was really impressed by what I was able to do for each project... I definitely felt more comfortable with coding in general after each project."

04

Discussion

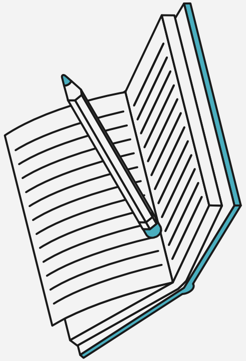


Discussion

The CS1-LLM course saw most students (79%) feeling capable of programming with LLMs, while a slight majority (59%) found LLMs helpful for learning. However, some students expressed concerns about over-reliance on the tool and gaps in fundamental skills. Despite these concerns, students created more advanced projects than typically expected in a CS1 course.

- **Student Performance:** Students performed similarly to past CS1 courses on code tracing and reading, but slightly lower on code writing tasks. However, their project scope and ability to solve large problems were impressive.

- **Essential Components:** Key elements include projects utilizing LLMs, teaching problem decomposition, testing, and incorporating LLM responses into class examples.



- **Course Changes:** Most changes involved LLM interaction, problem decomposition, and larger projects, though basic coding skills were still covered.

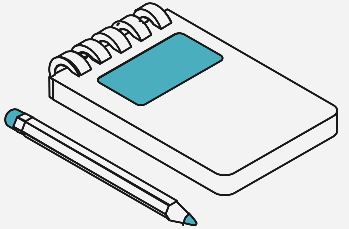
- **When to Introduce LLMs:** Introducing LLMs in the first week was challenging, so delaying their introduction slightly in future offerings is recommended.

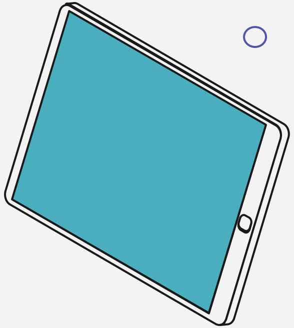
- **Clear Expectations:** Students were unclear about what they should accomplish with and without LLMs. Future courses will provide clearer guidance.

- **Learning Goals Alignment:** Assessments must be aligned with updated course goals, avoiding past questions that don't fit the new objectives.

- **Non-Determinism in Live Coding:** LLM responses varied across class sections, prompting the team to pre-paste Copilot responses into slides.

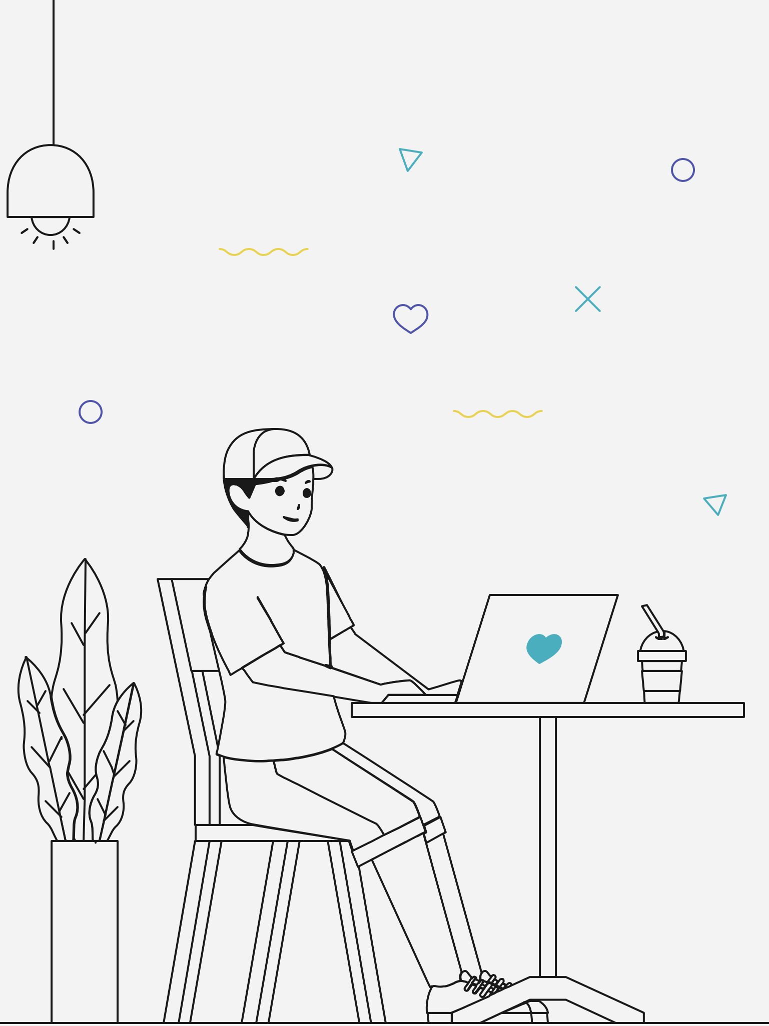
- **Login Issues in Exams:** Signing into GitHub for exams was time-consuming due to credential and verification issues. Future exams should allow extra time or personal device use for easier access.





05

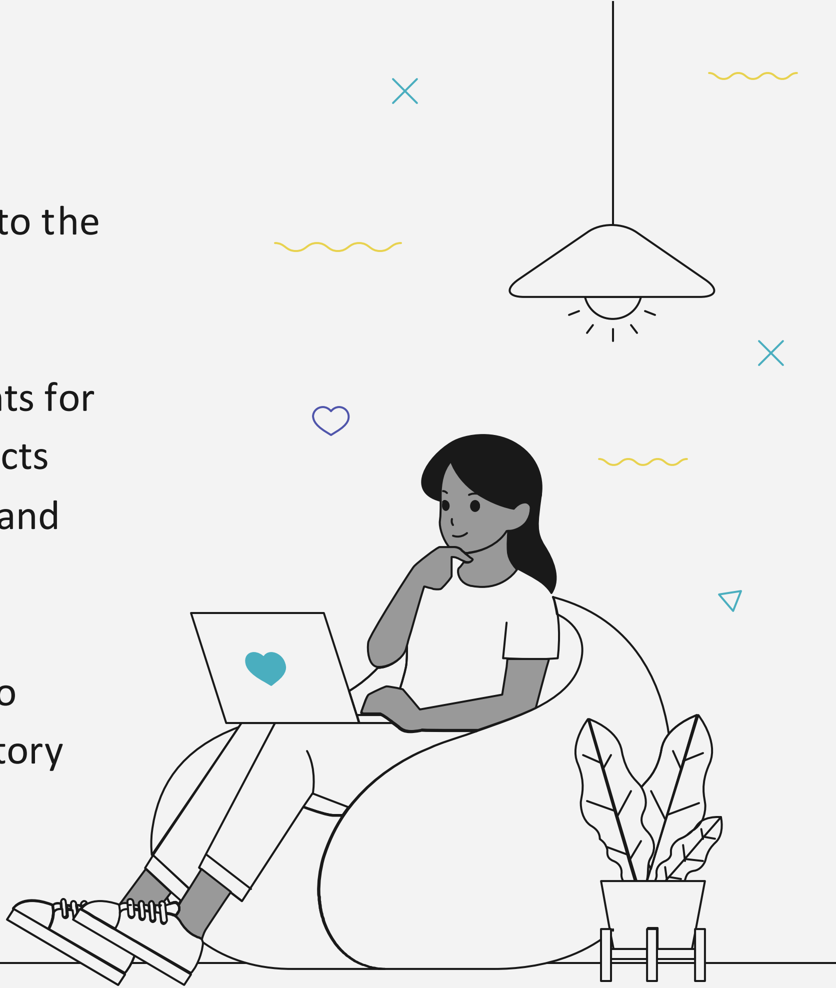
Conclusion



A new introductory programming course (CS1-LLM) was introduced, integrating large language models (LLMs) into the curriculum.

The course's revised learning goals, structure, and insights for future adopters were shared. Surveys and student projects revealed that students valued the open-ended projects and believed the LLMs helped their learning.

The course is seen as an initial step toward using LLMs to enhance student outcomes and experiences in introductory computer science education.



Thanks!