Exploring Student Behaviors and Motivations when using AI Teaching Assistants with Optional Guardrails

Amanpreet Kapoor[1], **Paul Denny**[2], Leo Porter[3], Stephen MacNeil[4], Marc Diaz[1]

University of Florida[1], University of Auckland[2], University of California San Diego[3], Temple University[4]

paul@cs.auckland.ac.nz

Motivation

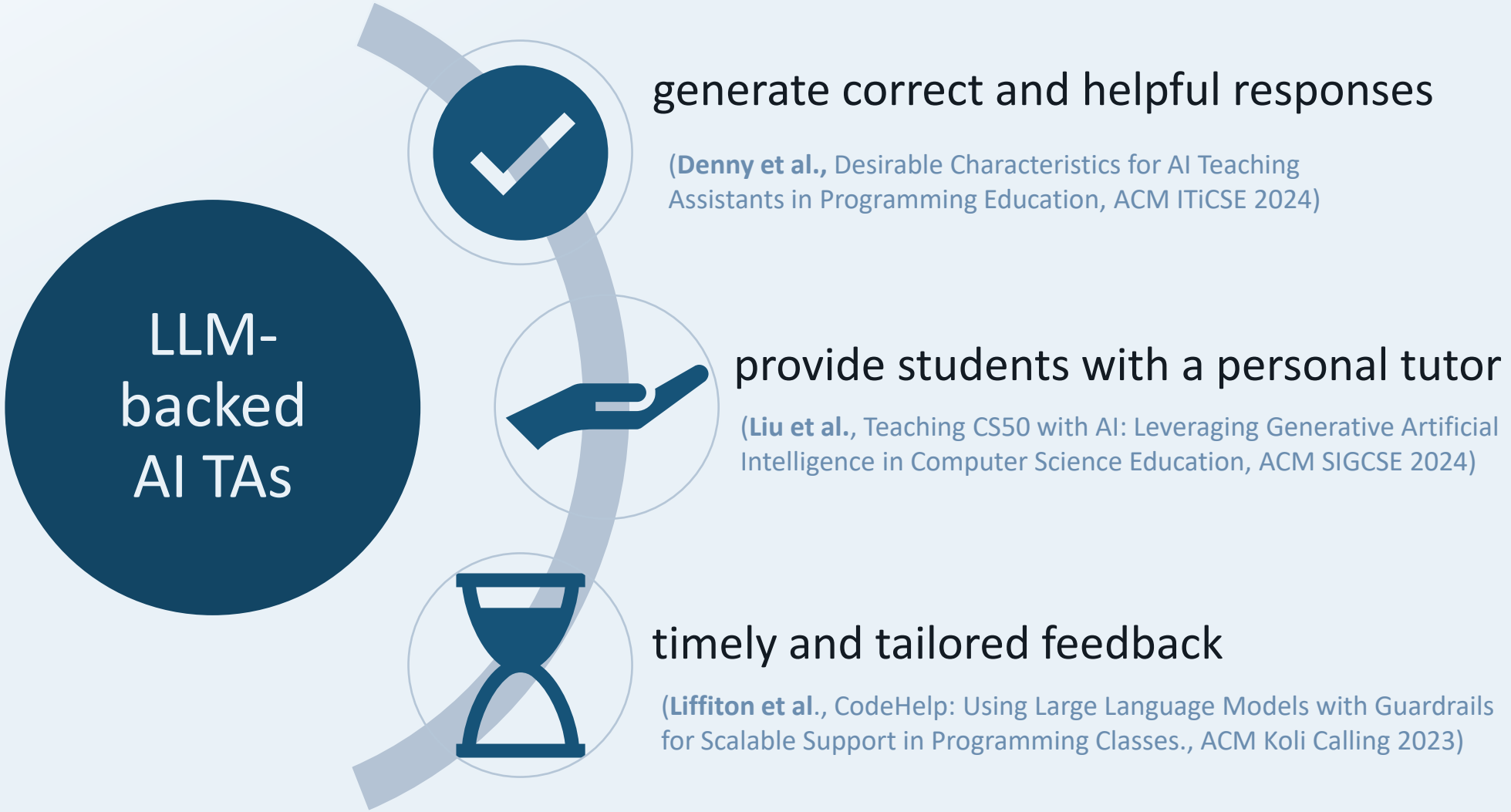
Source: Gemini

Providing timely and consistent human feedback in large introductory programming courses does not scale!

Source: Gemini

Solution: AI TA's backed by large language models (LLMs)

Recent Work on AI TAs: Benefits



generate correct and helpful responses

(**Denny et al.,** Desirable Characteristics for AI Teaching Assistants in Programming Education, ACM ITiCSE 2024)

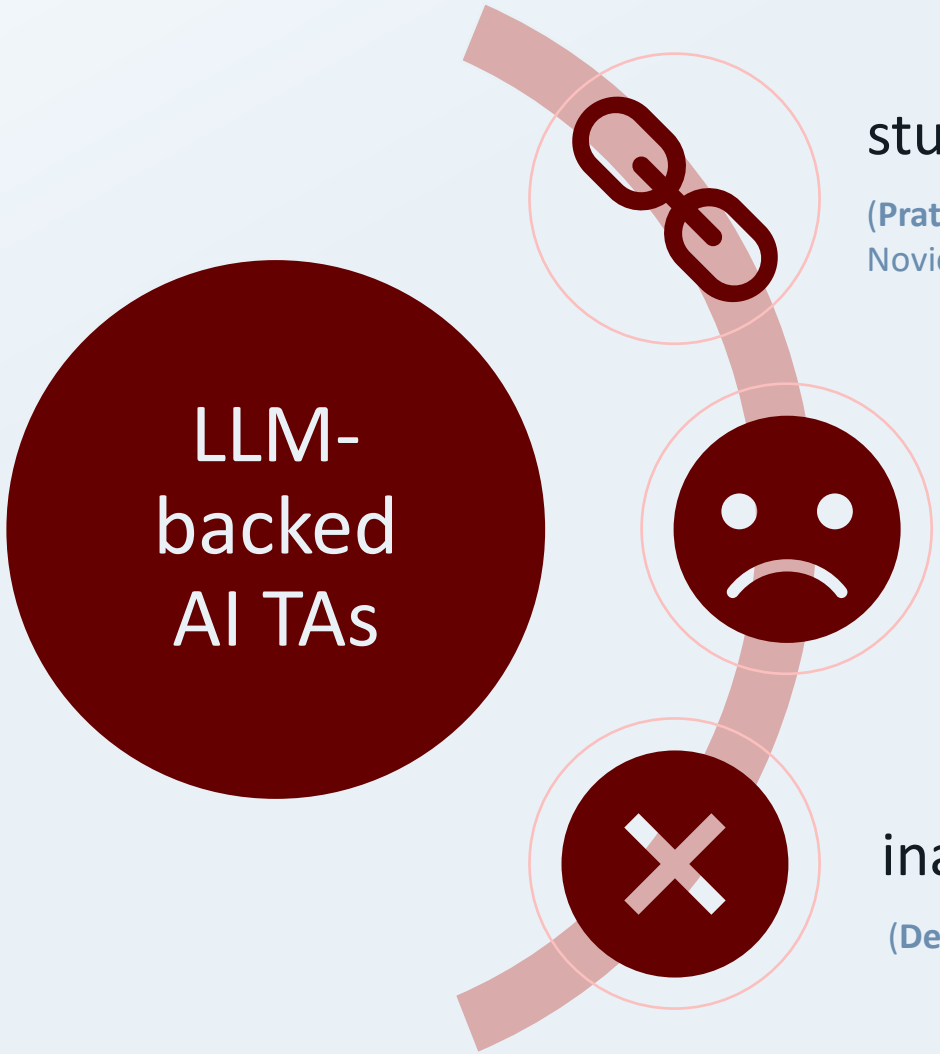
provide students with a personal tutor

(**Liu et al.**, Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education, ACM SIGCSE 2024)

timely and tailored feedback

(**Liffiton et al**., CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes., ACM Koli Calling 2023)

Recent Work on AI TAs: Drawbacks



student over-reliance leading to underdeveloped skills

(**Prather et al.,** The Widening Gap: The Benefits and Harms of Generative AI for Novice Programmers, ACM ICER 2024)

academic misconduct

(**Sheard et al.**, Instructor Perceptions of AI Code Generation Tools - A Multi-Institutional Interview Study., ACM SIGCSE 2024)

inaccuracies in the AI-generated content

(**Denny et al.,** Computing Education in the Era of Generative AI, ACM CACM 2024)

Recent Work on AI TAs: Adding Guardrails

CodeHelp

Code:

```
def character_remover(word, character):
    word = word.remove(character)
    return word
```

Error: `Attribute Error`

Issue / Question: I want this function to print the word, excluding the character.

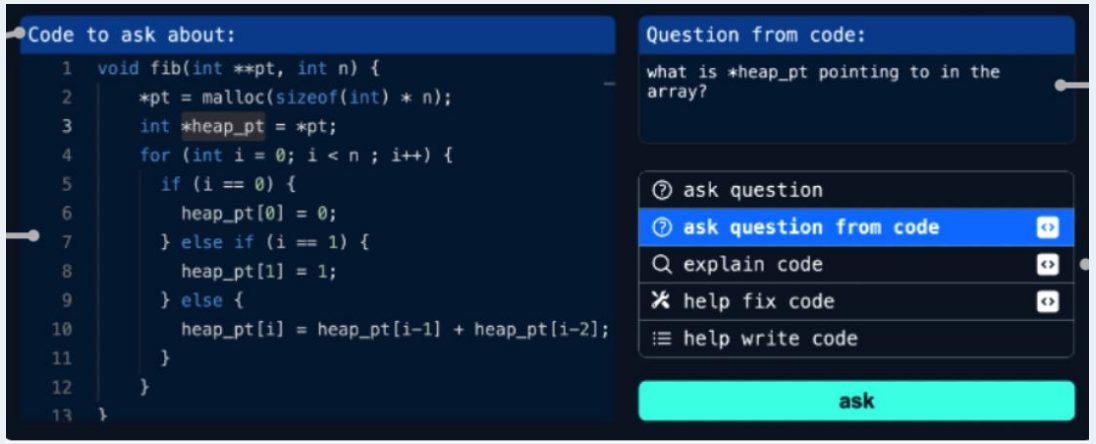
CodeHelp's Response: The error message you are seeing is an Attribute Error, which means that the object you are trying to access does not have the attribute or method you are trying to use. In this case, the object is the string `word` and the attribute you are trying to use is `remove()`. The `remove()` method is not a valid method for strings, so you are getting this error.

You can use the `replace()` method instead. This method takes two parameters: the substring you want to replace and the substring you want to replace it with. In this case, you want to replace the character with an empty string. To do this, you can pass an empty string as the second parameter to the `replace()` method. This will remove the character from the string.

For example, if you have a string `word = 'Hello'` and a character `character = 'l'`, you can remove the character from the string by calling `word.replace(character, '')`. This will return a new string with the character removed, `'Heo'`.

(**Liffiton et al**., CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes., ACM Koli Calling 2023)

CodeAid

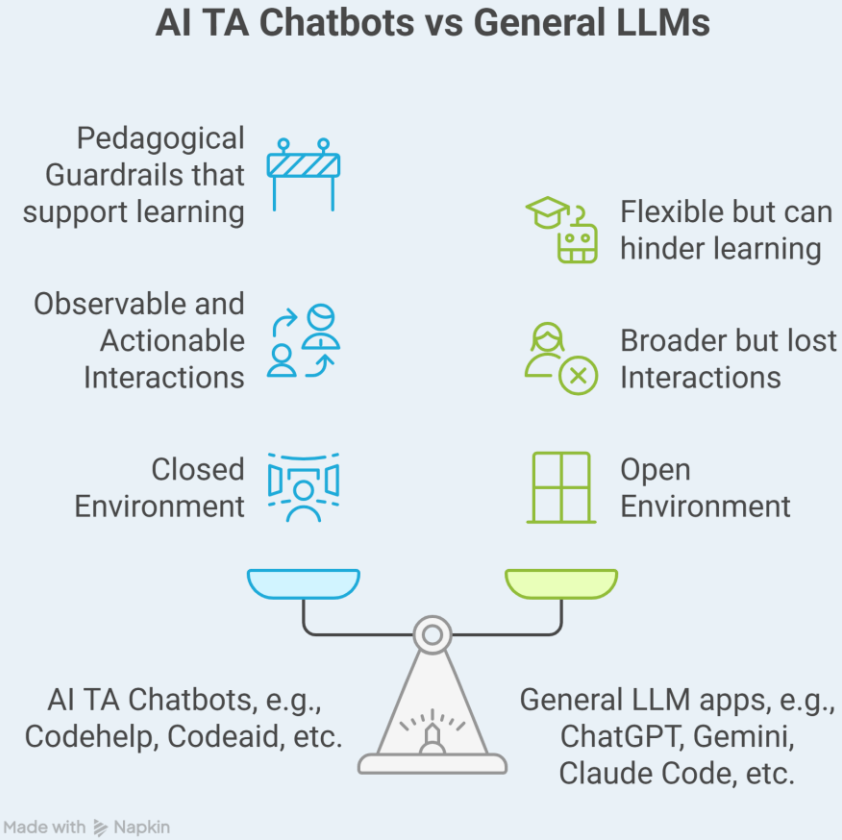


(**Kazemitabaar et al**., CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs, ACM CHI 2024)

Recent Work on AI TAs: Adding Guardrails

Source: Gemini

Problem: Overly Restrictive Guardrails can promote general LLM applications



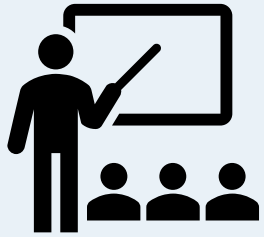
Source: Napkin.ai

General LLMs can lose learning interactions and impede learning or promote over-reliance

Study Context

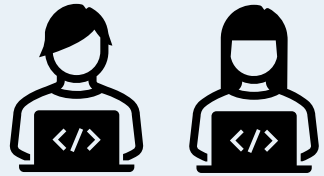
Observational study

Fall 2024



Introductory programming course

Taught by Prof. Paul Denny @ University of Auckland
12-weeks semester, # students=1,034,
Corpus for analysis, N=885

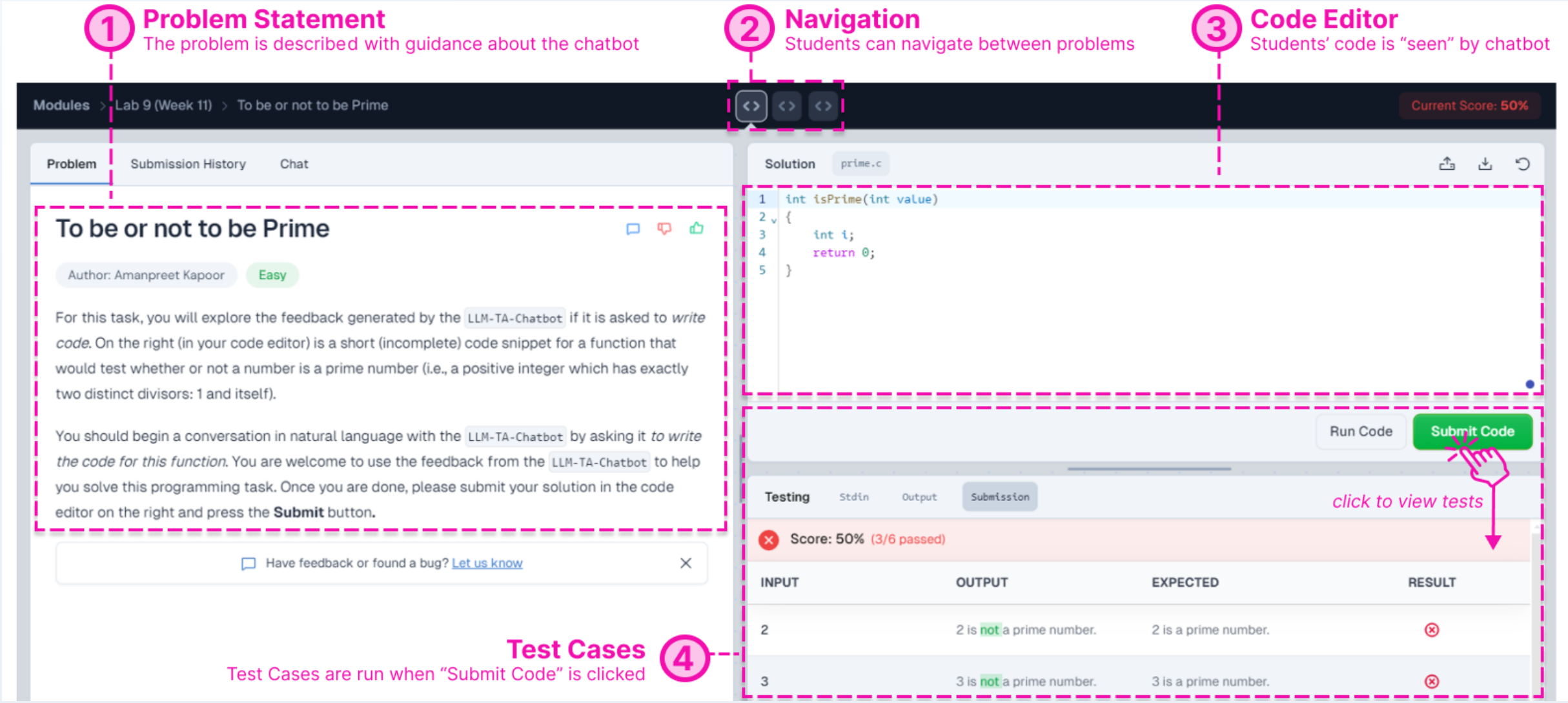


Lab 9 (C Language)

- 1 out of 10 marks (0.1% of course grade)
- One code writing + Two code debugging tasks
- Two quantitative + Two qualitative questions
- Ran over nine days

Study Context: Tool

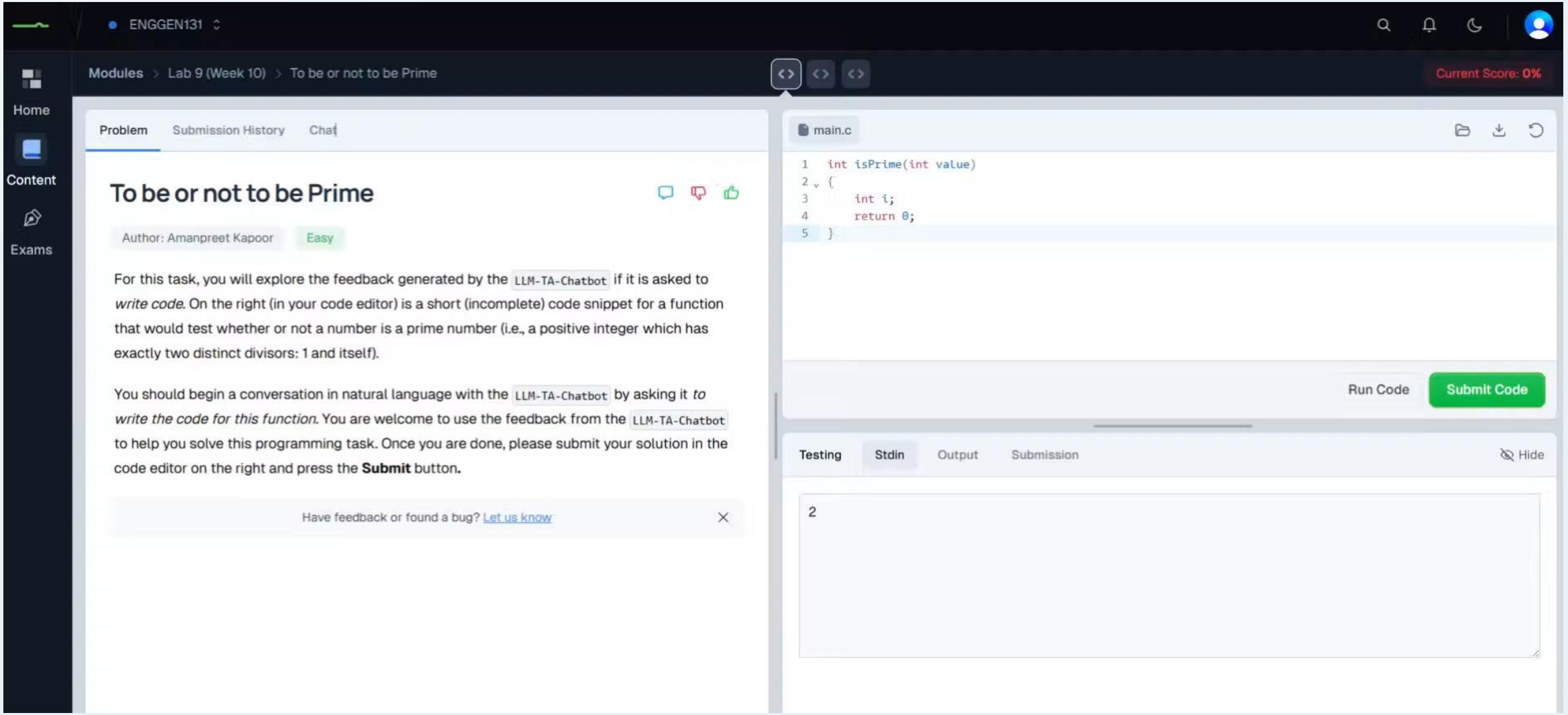
Edugator



Interface for Edugator Tool, https://edugator.app/

Study Context: Tool

Edugator



Testing solution and AI chat interface in Edugator to ask questions and solicit solutions

Study Context: Tool Implementation

Implementation of AI TA Chatbot

AI TA with Guardrails: using Prompt (GPT4o)

"Respond to the student with a **brief educational explanation**, **helping the student figure out the issue** and **understand what they're doing incorrectly**. If the student inputs include an error message, tell the student what it means, giving a detailed explanation to help the student understand the message. […]. Be **positive and encouraging**, and keep it **conversational**, meaning try to push the student in the right direction before outright explaining everything. If the **student's issue requests code**, **tell them you cannot provide any code**. [...]"

'See Solution': no Guardrails (GPT4o)

"Your goal is to provide a **detailed, educational explanation of the problem**, including the **correct code structure and logic**. Your responses should be concise, clear, and easy to understand. Ensure that the **solution is accurate**, follows best practices for the given programming language, and leverages the provided template code. [...]"

Context:

Prompt

Chat History

Course Language

Current Student Solution

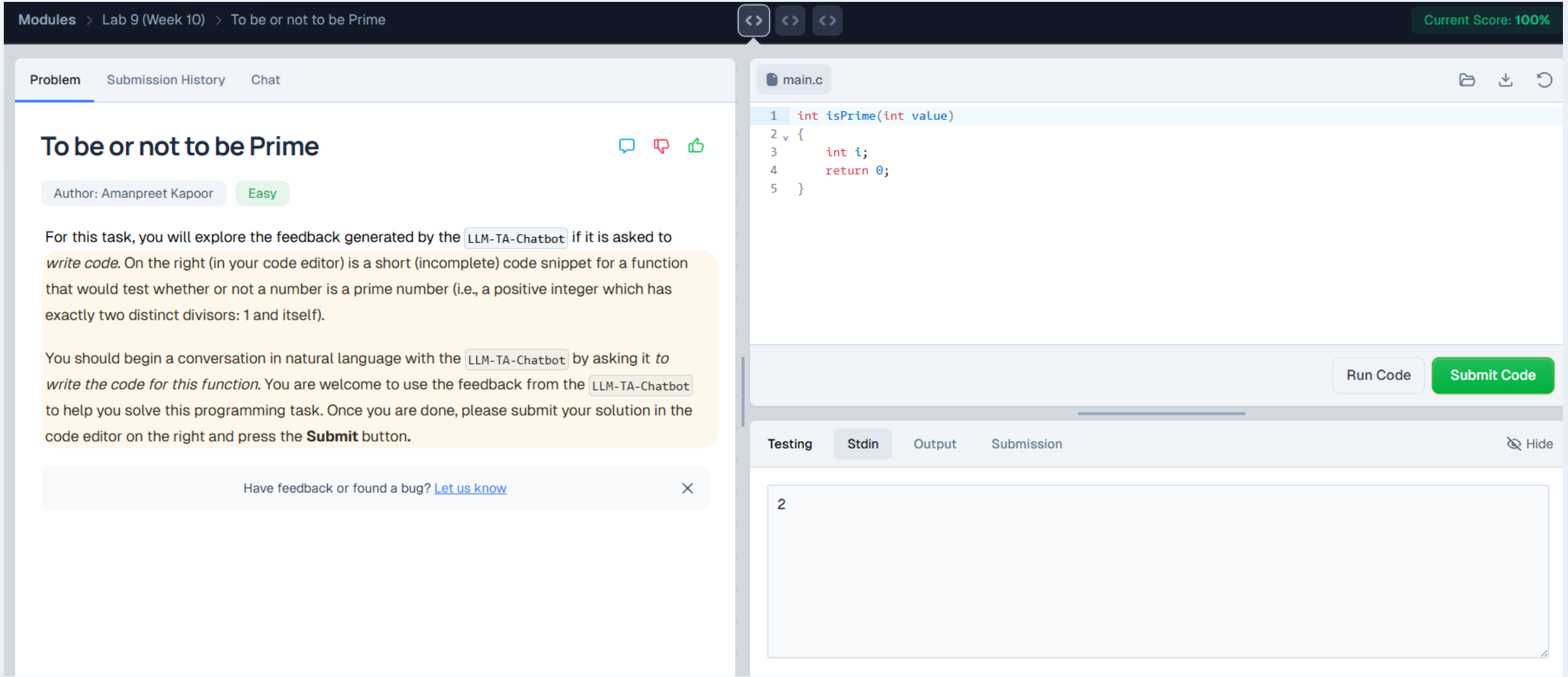
+

Problem Description

Study Context

- One code writing and two code debugging tasks focused on nested loops and two-dimensional arrays.
- No penalty for using ‘See Solution’ button.
- Lab Handout stated: “ The chatbot will respond to your questions helping you approach a problem without giving you the solutions. In case you want to see potential solution code, you can click ‘See Solution’ ”.
- AI policy in course:
  - Students were discouraged from using tools like ChatGPT
  - Custom AI-powered teaching tools like Codehelp, Prompt Programming, Edugator, etc. were allowed.

Activity 1, IsPrime (Code Writing)



Activity 2, IsRepeated (Code Debugging)

Activity 3, SurroundingSum (Code Debugging)

Modules > Lab 9 (Week 10) > Surrounding Sum

Current Score: 50%

Problem | Submission History | Chat

Surrounding Sum

Author: Amanpreet Kapoor | Easy

The code for the function, `SurroundingSum()` is on the right (in your code editor), however it contains a bug. The `SurroundingSum()` function returns the sum of numbers surrounding the number specified by a `row` and `col` as shown in the image below.

int sum = SurroundingSum(values, 4, 4);

	0	1	2	3	4	5	6	7	8	9
0	1	5	3	9	7	6	2	10	4	8
1	8	1	4	6	2	7	9	3	5	10
2	1	7	3	2	1	4	6	5	0	2
3	0	1	7	3	2	1	4	6	5	0
4	4	0	1	7	3	2	1	4	6	5
5	8	4	0	1	7	3	2	1	4	6
6	1	8	4	0	1	7	3	2	1	4
7	4	3	10	8	5	9	2	7	6	1
8	6	2	8	7	1	4	10	9	3	5

main.c

```
1  int SurroundingSum(int values[10][10], int row, int col) {
2      int i, j;
3      int sum = 0;
4      for (i = row - 2; i <= row + 2; i++) {
5          for (j = col - 2; j <= col + 2; j++) {
6              if (i != row || j != col) {
7                  sum += values[i][j];
8              }
9          }
10     }
11     return sum;
12 }
```

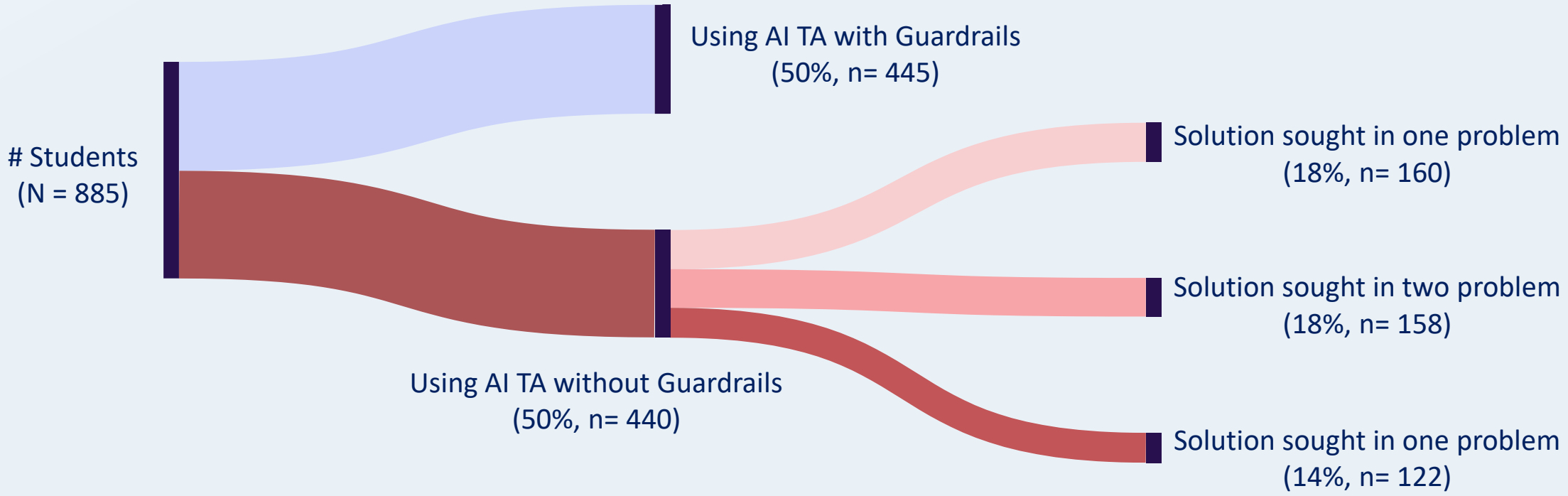
Run Code | Submit Code

Testing | Stdin | Output | Submission | Hide

```
4 4
1 5 3 9 7 6 2 10 4 8
8 1 4 6 2 7 9 3 5 10
1 7 3 2 1 4 6 5 0 2
0 1 7 3 2 1 4 6 5 0
4 0 1 7 3 2 1 4 6 5
8 4 0 1 7 3 2 1 4 6
1 8 4 0 1 7 3 2 1 4
4 3 10 8 5 9 2 7 6 1
6 2 8 7 1 4 10 9 3 5
```

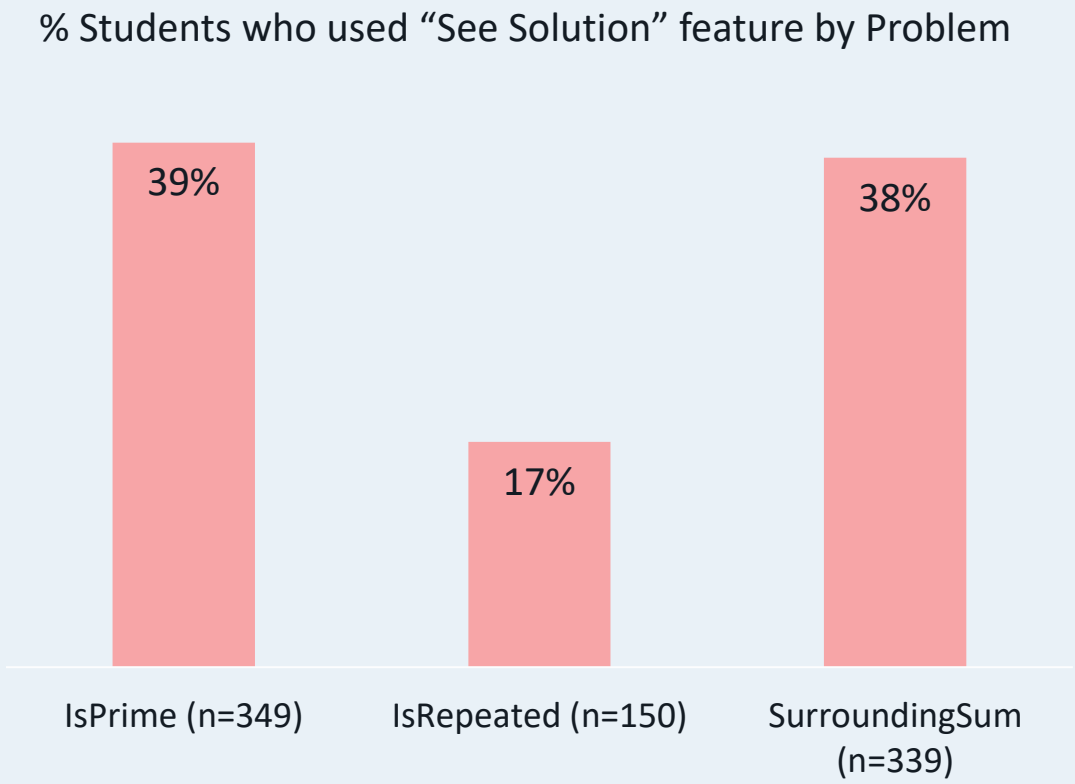
Findings (RQ1a)

RQ1a. To what extent is the "See Solution" feature used, and how does this usage relate to student performance in the course**?**



Findings (RQ1a)

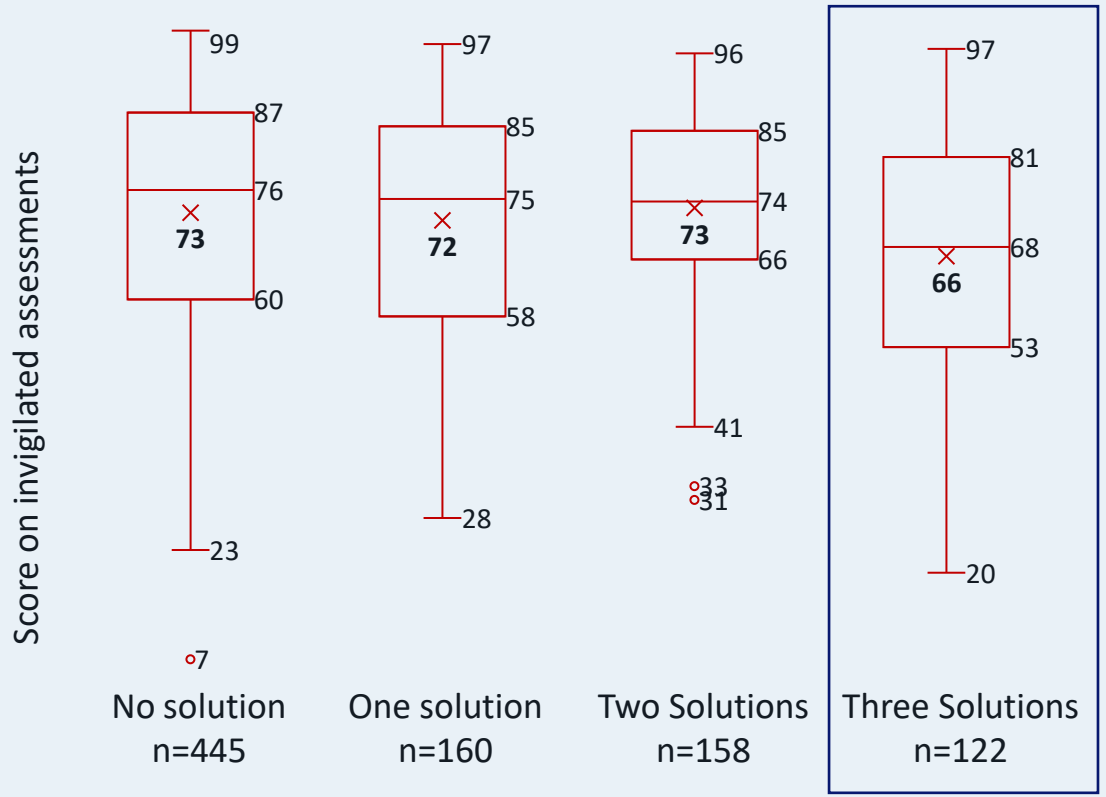
RQ1a. To what extent is the "See Solution" feature used, and how does this usage relate to student performance in the course**?**



Findings (RQ1a)

RQ1a. To what extent is the “See Solution” feature used, and **how does this usage relate to student performance in the course?**

Invigilated Score distribution of Students across Solution Seeking (N=885)



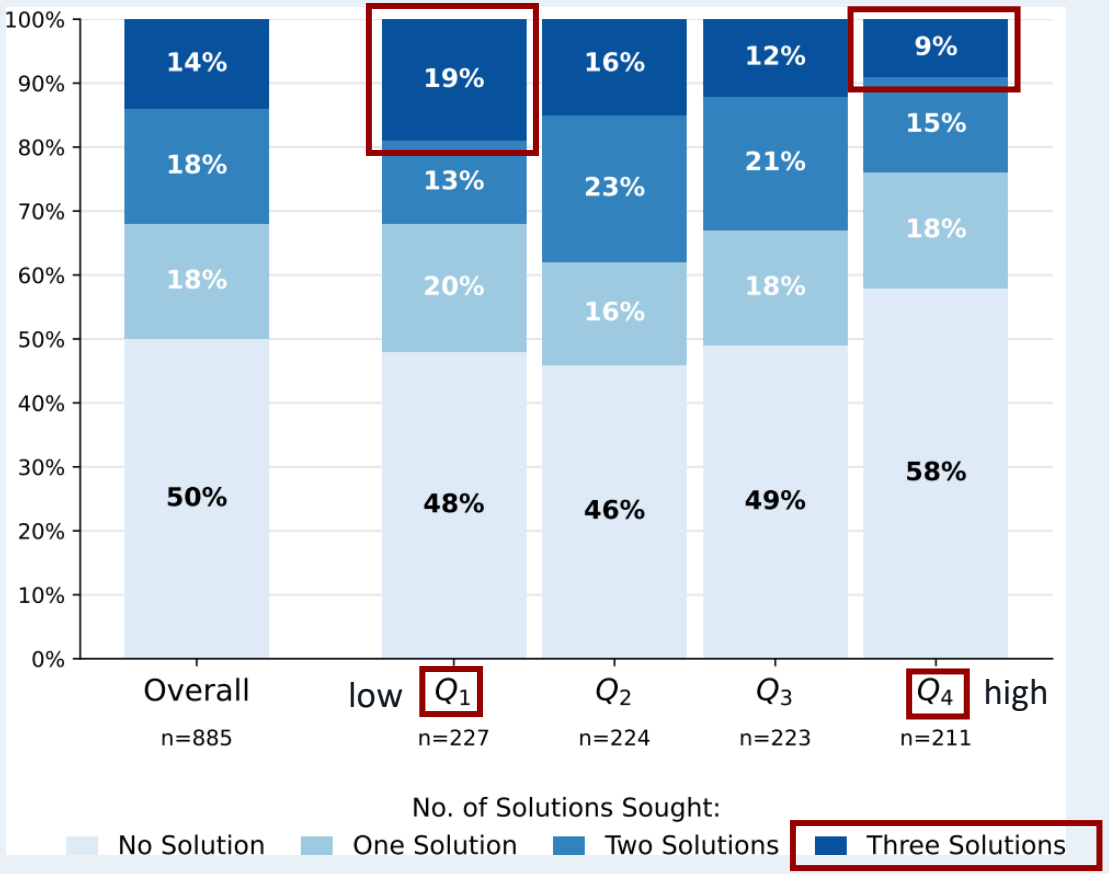
Students who used “See Solution” in all problems tend to score lower.

Kruskal-Wallis H test,
χ^2 (3) = 14.4, **p = .002***

Findings (RQ1a)

RQ1a. To what extent is the "See Solution" feature used, and **how does this usage relate to student performance in the course?**

Usage of "See Solution" feature by course performance quartile (Q_1=low, Q_4=high)

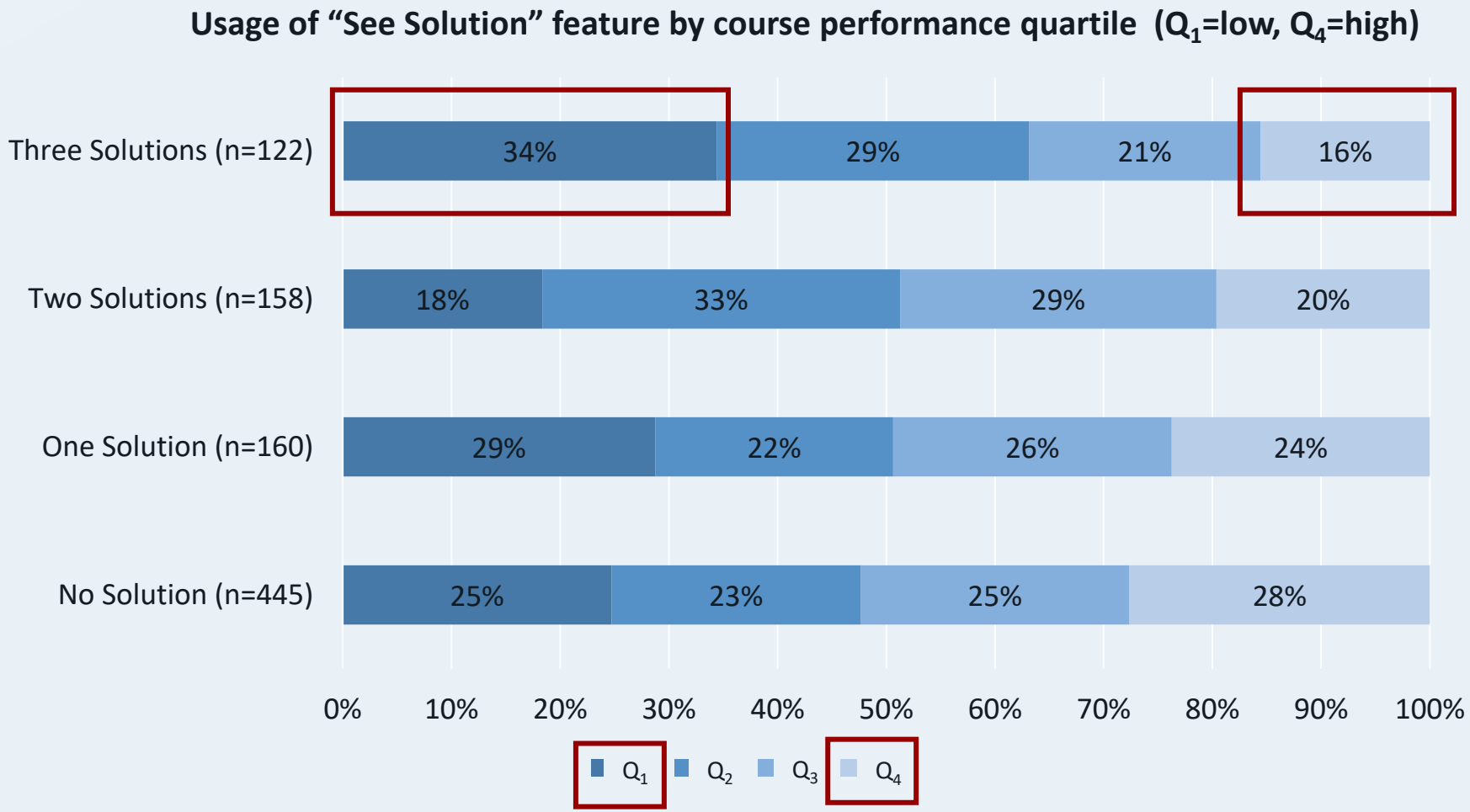


Students in **Q_4** were more likely to **not use the feature** and were **less likely to use it** in all **three problems**.

For instance, a student in Q_1 was twice as likely to use it in all problems than a student in Q_4.

Findings (RQ1a)

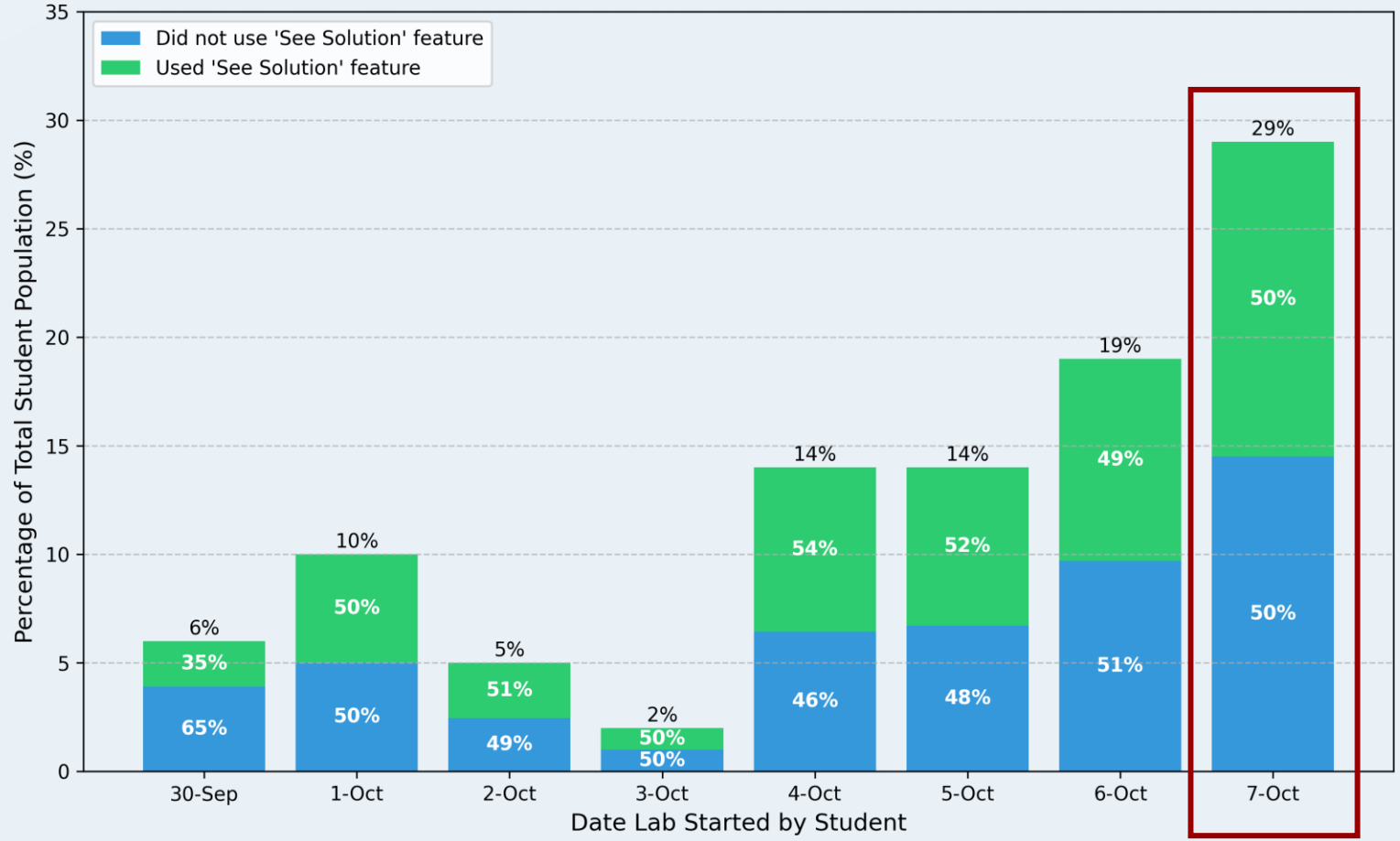
RQ1a. To what extent is the “See Solution” feature used, and **how does this usage relate to student performance in the course?**



Findings (RQ1b)

RQ1b. **How does the timing of student engagement with the lab tasks relate to the use of the "See Solution" feature?**

Usage of "See Solution" feature by date based of when a student started the lab (Deadline: 7-Oct, N = 883)



Decision to use the "See Solution" feature was independent of when they started the lab.

chi-square test of independence
χ^2 (7) = 5.92, p = 0.55

Findings (RQ1b)

RQ1b. **How does the timing of student engagement with the lab tasks relate to the use of the "See Solution" feature?**

Date	% Submissions (N=13024) Q_1	Q_2	Q_3	Q_4	% Solution Requests (N=1049) Q_1	Q_2	Q_3	Q_4
29-Sep	0%	0%	0%	0%	0%	0%	0%	0%
30-Sep	2%	5%	5%	13%	0%	3%	3%	14%
1-Oct	3%	5%	13%	14%	5%	3%	19%	10%
2-Oct	1%	10%	4%	7%	4%	5%	4%	15%
3-Oct	2%	3%	3%	4%	1%	3%	2%	4%
4-Oct	9%	10%	14%	21%	11%	12%	15%	22%
5-Oct	13%	12%	20%	15%	14%	13%	17%	12%
6-Oct	19%	20%	21%	15%	14%	27%	16%	16%
7-Oct	52%	35%	20%	11%	52%	34%	26%	8%

Heatmap of "See Solution" feature usage rate and submission rate w.r.t. student performance (N=884, Q_1=low, Q_4=high)

- Low-performing students procrastinate more compared to high-performing students.

- Usage of "See Solution" feature appears largely independent of procrastination, although procrastination is related to student performance.

Findings (RQ2)

RQ2. **What factors motivate students to use or refrain from using the "See Solution" feature when engaging with an AI TA?**

Method:

Open-ended question after they completed the three lab tasks – 839 responses.

Question:

If you used the "See Solution" feature to generate a code solution for any of the three problems, **explain your rationale for using this feature**. Reflect on the usefulness of this feature and the extent you used the generated solution in your final submission for the respective problem. Alternatively, if you did not use this "See Solution" feature, comment on why you didn't use it.

Analysis:

- Reflexive thematic analysis approach by a single researcher (**Braun & Clarke**, Using thematic analysis in psychology, Qualitative Research in Psychology, 2006)
- Researcher actively and iteratively constructed their understanding of the responses, rather than seeking an objective 'truth'

Findings (RQ2)

RQ2. **What factors motivate students to use or refrain from using the "See Solution" feature when engaging with an AI TA?**

Reasons not to view solutions:

1. Perceived learning value and sense of accomplishment

intrinsic value associated with independently solving the problems.

*"I did not use the see solution tool because I find that when you see an already completed solution, it **takes away the learning experience** that comes from crafting a unique solution yourself".* – P36

2. Not needing help

guardrailed chatbot already provided sufficient support.

*"I did not use the see solution feature as I **felt the chat bot offered more than enough assistance** for each of the problems".* – P137

Findings (RQ2)

RQ2. **What factors motivate students to use or refrain from using the "See Solution" feature when engaging with an AI TA?**

Reasons not to view solutions:

3. Ethical concerns

described using the solutions as cheating.

*"I chose not to use the 'See Solution' feature as it **felt a bit like cheating** and I wanted to figure it out on my own. Overall, once I learnt how to ask the [chatbot] the right questions, I was able to figure it out much more easily".* – P400

Findings (RQ2)

RQ2. **What factors motivate students to use or refrain from using the "See Solution" feature when engaging with an AI TA?**

Reasons to view solutions:

1. Problem-solving assistance

seek help when stuck or to verify work.

*"... after asking the bot 5 to 6 questions, it does give me an opportunity **to check and see if my thought process** was indeed **in the right direction**. I would definitely say that using the solution straight away would defeat the purpose of such a learning platform".* – P801

2. Time pressure

tight deadlines and competing priorities.

*"I am submitting this lab quite late so I **did not really have time** to completely rewrite and debug the code myself so I clicked see solution. [However, without the time pressure] I would genuinely give it a go before wanting to peak at the see solution option".* – P277

Findings (RQ2)

RQ2. **What factors motivate students to use or refrain from using the "See Solution" feature when engaging with an AI TA?**

Reasons to view solutions:

3. Lack of self-regulation skills

lack of self-control when it came to using the solution feature.

*"I honestly **just got lazy**, but when it was more straight forward I would not use [the 'See Solution'] feature".* – P941

8.4% of 440 students who used the feature, clicked the button after solving the problem

4. Curiosity

used the solution to assess its accuracy or to compare their correct solution to the tool's solution.

*"I used the 'see solution' feature after I'd done the problem, simply to **compare my solution with a model one**. I'd followed the advice of the AI bot so the solutions looked pretty much the same".* – P975

Discussion & Conclusion

- **Guardrail paradox (perceived value vs. actual use):** Although students reported valuing the learning-oriented AI TA with guardrails[1], many bypassed it when given the option; approximately 50% of students used the “See Solution” feature.

- **Motivations for bypassing guardrails**: Students cited psychosocial factors such as limited self-regulation and time pressure - patterns similar to those observed in interactions with human TAs[4] - as well as reasons that may support learning (e.g., verifying answers or exploring alternative solutions).

- **Performance-based patterns:** Low-performing students used the “See Solution” more for all tasks and exhibited greater procrastination, consistent with prior work[2,3].

References

1. **Denny et al.,** Desirable Characteristics for AI Teaching Assistants in Programming Education, ACM ITiCSE 2024
2. **Liao et al.,** Behaviors of Higher and Lower Performing Students in CS1, ACM ITiCSE 2019
3. **Zhang et al**., Exploring the Impact of Voluntary Practice and Procrastination in an Introductory Programming Course, ACM SIGCSE 2022
4. **Lim et al.,** Student Expectations of Tutors in Computing Courses ACM SIGCSE 2023

Implications for Research and AI TA Design

- **Early risk identification:** Persistent reliance on solution features especially among low-performing students may signal self-regulation challenges and could be used to identify at-risk students.

- **Pedagogical flexibility:** Providing optional access to alternative solutions can support learning in low-stakes or formative contexts, with instructors able to enable or disable guardrails at the problem level in systems.

- **Open question**: When should the student have access to the solution for high stakes problem?

Study Limitations

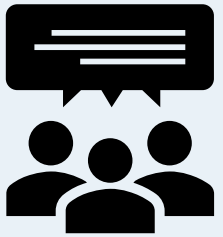
- **Observational Inquiry:** Students were not randomly assigned to conditions and results reflect correlation, not causation.
- **Contextual Specificity:** Conducted in a single lab at one research university; results may not generalize to other academic environments.
- **Design & Tooling Bias:** The AI TA lacked problem descriptions in its prompts, leading to limited guidance that may have pushed frustrated students toward the “See Solution” button.
- **Uncontrolled External Resources**: Extended access could allow peer collaboration or external AI use (e.g., ChatGPT); however, the low-stakes nature of the lab (0.1% of course grade) and a penalty-free “See Solution” option may have reduced incentives for misconduct.

Acknowledgements

National Science Foundation Award #2417374

Edugator

Dustin Karp and Prayuj Tuli for implementing the “See Solution” feature



Reviewers

Questions?

paul@cs.auckland.ac.nz
kapooramanpreet@ufl.edu