Thinking Before Prompting: Imparting Responsible LLM Use and Security Awareness in a CS Problem-Solving Course

Dr. Sehrish Basir Nizamani, Virginia Polytechnic Institute and State University

Sehrish Basir Nizamani is a Collegiate Assistant Professor in the Department of Computer Science and a Faculty Affiliate with the Center for Human-Computer Interaction and the Center for Future Work Places and Practices at Virginia Tech. Her research lies at the intersection of Human-Computer Interaction and Digital Education. She is particularly interested in how Large Language Models (LLMs) are reshaping both education and interaction design. Her current work explores the integration of LLMs into undergraduate computing courses to foster critical thinking, ethical reflection, and collaborative problem-solving. In parallel, she investigates how LLMs can be applied across various HCI phases to enhance interface design and user experiences. Her ongoing projects also examine the use of LLMs to evaluate classroom dynamics, focusing on posture, spatial layout, and engagement in smart learning environments.

Nikitha Donekal Chandrashekar, Virginia Polytechnic Institute and State University

I am a Postdoctoral Associate in Virginia Tech Carilion Health Systems. My research interests are focused on integrating emerging technologies like XR and Generative AI to help with professional skill training and Education. I am also committed to teaching and pedagogy. During my doctoral studies, I served as an Instructor for CS 2114: Data Structures and Software Design for two years. I am now working with the Computer Science Department to help include LLM to the curriculum.

Ms. Margaret O'Neil Ellis, Virginia Tech

Professor of Practice, Computer Science Department, Virginia Tech

Naren Ramakrishnan, Virginia Polytechnic Institute and State University

Naren Ramakrishnan is the Thomas L. Phillips Professor in the Department of Computer Science at Virginia Tech. He is the director of the Virginia Tech's Sanghani Center for Artificial Intelligence and Data Analytics and the Institute for Advanced Computing lead for AI and machine learning. Naren is a Fellow of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronic Engineers (IEEE), and the American Association for the Advancement of Science (AAAS).

Thinking Before Prompting: Imparting Responsible LLM Use and Security Awareness in a CS Problem-Solving Course

Sehrish Basir Nizamani
sehrishbasir@vt.edu
Department of Computer Science
Virginia Tech

Nikitha Donekal Chandrashekar
nikitha@vt.edu
Department of Computer Science
Virginia Tech

Margaret Ellis
maellis1@vt.edu
Department of Computer Science
Virginia Tech

Naren Ramakrishnan
naren@vt.edu
Department of Computer Science
Virginia Tech

July 8, 2026

Abstract

Large Language Models (LLMs) have rapidly become tools of fascination for students and professionals alike, offering a sense of effortless progress, almost like a time machine that transports users from an initial idea to a polished state of completion within moments. Their convenience, however, often obscures underlying risks: in the rush to achieve quick results, users may unknowingly share sensitive, private, or protected information with these systems or generate content that compromises privacy and security. While LLMs offer immense potential for creativity, productivity, and problem-solving, their secure and responsible use requires awareness and caution.

We redesigned a Computer Science (CS) Problem solving course to integrate Artificial Intelligence (AI); this paper discusses how we addressed ethical, privacy, and security implications of using LLMs throughout the course. During Week 2, students explored the risks and benefits of LLMs and engaged with materials on LLM privacy and safety, how the government and individual organizations are addressing privacy and security needs[1]. In Week 3, students completed the CS Personality Bot Interaction and Ethics Assignment, where they interacted with Character AI and reflected on ethical concerns through assigned readings from CBS News and NPR about lawsuits involving minors using chatbots. They connected these discussions to the ACM Code of Ethics, focusing on professional responsibility, respect for user privacy, and harm prevention. By Week 5, the focus shifted to Software Engineering, where students analyzed how LLMs intersect with security practices. Content on security concerns, secure coding practices, how LLMs can assist with security, and security risks of LLMs, encourage students to consider both the potential and the vulnerabilities of integrating AI tools

into software development. Collectively, this multi-week intervention fostered ethical and security awareness while promoting responsible AI use in professional computing contexts.

As part of the intervention, we conducted a pre and post course survey which included an important question: ”What are the security concerns around LLMs?” This question was designed to capture students’ baseline awareness before the instructional modules and measure conceptual growth afterward. The qualitative analysis of the open ended responses showed an increase in the students’ recognition of key LLM-related security issues, particularly in areas such as data leakage, personal information gathering, and malicious use. We also noted the emergence of new codes in their answers such as ”hallucination”, ”privacy issues” and ”prompt injection attacks”. These additions indicate that students developed a more nuanced and technically informed understanding of LLM-related risks, extending beyond general privacy and misuse concerns to encompass deeper issues of model behavior and reliability. This shift suggests that the course intervention effectively deepened students’ understanding of real-world AI security and ethical risks. Our results demonstrate that this lightweight approach of highlighting potential LLM security risks in an introductory computing course can increase students’ awareness of the need for responsible usage.

Introduction

Large language models have become part of everyday practice across education, professional work, and personal activities. Education plays an important role in preparing students to engage with emerging AI technologies by developing the skills and competencies needed for future academic and professional contexts[2]. LLMs can be used in programming education for feedback, debugging, repair and personalized learning[3]. Their use is often motivated by clear benefits, including faster task completion and improved accuracy for novice programmers[4]. Despite their advantages, LLMs introduce important security and privacy risks that require careful consideration, including vulnerabilities such as prompt attacks, data poisoning, and information leakage[5].

Students may share sensitive information during interactions with these systems without fully considering where that information is stored or how it is used. They may rely on generated outputs that are incorrect or misleading, or remain unaware of vulnerabilities such as data leakage, hallucinated responses, or adversarial manipulation. As LLM use becomes more common in undergraduate courses, particularly at the introductory level, questions arise about how students come to understand these risks and how they develop habits around responsible use. Focusing only on functional proficiency is increasingly insufficient in contexts where AI tools are woven into everyday learning practices.

Within educational settings, students represent a particularly important group to consider. They are often early adopters of new technologies and frequent experimenters, and they are still forming assumptions about professional norms and acceptable practices. In the absence of explicit guidance, students may develop patterns of LLM use that prioritize convenience over caution. Such practices can persist beyond the classroom and shape how students engage with AI tools in later academic or professional environments. When AI systems become integrated into real-world applications, security and privacy failures can have serious consequences, including

insecure software generation, information leakage, and threats to system integrity[6]. For this reason, attention to responsible and secure LLM use should not be treated as an optional addition to computing education, but as part of preparing students for sustained engagement with AI enabled systems.

In this paper, we explore how awareness of LLM security, privacy, and ethical concerns can be introduced early through integration into existing coursework. We present a case study of a Computer Science (CS) problem solving course in which discussions of LLM risks and responsibilities were embedded across multiple weeks, alongside students' regular use of these tools. Rather than positioning security as a separate or advanced topic, the course introduced these considerations in relation to everyday learning activities. As illustrated in Figure 1, security related topics were introduced progressively, beginning with general awareness and extending to ethical reflection and software engineering contexts.

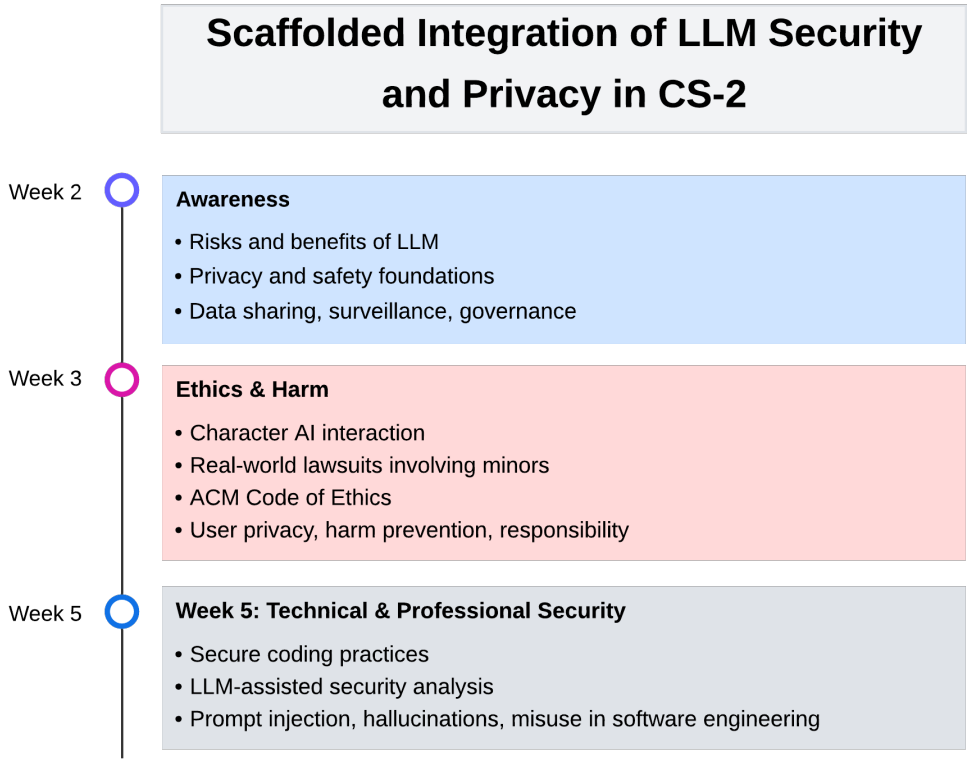

Figure 1: Scaffolded integration of LLM security topics across a CS problem-solving course.

To examine how students engaged with these ideas, we analyze responses to an open ended survey question administered before and after the course, which asked students to describe security concerns related to LLMs. This qualitative analysis allows us to consider not only whether students became more aware of security issues, but how their understanding shifted over time. The responses suggest movement from broad or uncertain concerns toward more specific descriptions of risk, including data leakage, privacy exposure, hallucinations, and prompt

injection attacks. These patterns point to the ways in which even limited instructional attention may support more informed reasoning about LLM security and responsible use in introductory computing contexts.

This work makes several contributions. First, it offers an example of how discussions of LLM security, privacy, and ethics can be integrated into an introductory computing course without displacing core technical material. Second, it provides empirical insight into how students' understanding of LLM related risks develops when they are given structured opportunities for reflection. Finally, it highlights the importance of early, user centered instruction in shaping how students engage with AI tools they already use. Together, these contributions add to ongoing conversations about responsible AI practice in computing education.

The instructional intervention described in this paper was part of a broader effort to introduce and normalize the use of AI tools within the course. Prior work has documented the design and early implementation of this course redesign, which aimed to help students engage with AI systems as part of their regular problem solving practice rather than as isolated or optional tools[7]. Building on this foundation, the present study focuses specifically on how issues of security, privacy, and responsible use were introduced and taken up by students over the course of the semester.

Literature Review

Widespread and informal use of LLMs has increasingly raised questions about how responsibility and security should be understood and addressed. Prior research suggests that LLMs are being taken up quickly by both students and instructors, often early in the curriculum and with little formal guidance. Across higher education, LLMs are increasingly being integrated into teaching, learning, assessment, research, and academic support activities[8,9]. This early use, however, often takes place without sustained attention to security, privacy, or ethical considerations, particularly in introductory computing courses. Related institutional efforts have applied similar frameworks to restructure individual courses around demystifying, using, and reflecting on LLMs, helping students build a working understanding of how these systems function before relying on them[10].

A recurring theme in the literature on LLM use in computing and software development is a focus on productivity. Empirical studies often describe LLMs as tools that support faster coding, improve developer productivity, and help streamline development workflows[11,12]. In educational contexts, these tools are commonly used to provide rapid feedback, assist with debugging, and support iterative learning, which can make it easier for students to complete tasks efficiently. While these uses are well documented, much of the literature centers on efficiency and convenience, with less attention to how such workflows may shape students' awareness of security, privacy, and reliability concerns, particularly for novice users.

At the same time, an expanding body of work has described security and privacy risks associated with LLMs, including data leakage, exposure of sensitive information, hallucinations, and prompt injection attacks[13,6]. As a result, these risks are most often discussed at the system level and remain somewhat removed from the everyday experiences of students, who typically encounter LLMs as end users rather than as designers or security specialists. For example, over-reliance on

LLM generated code can result in insecure patterns or vulnerability without students even recognizing it.

When security considerations are addressed in educational settings, they are often introduced later in the curriculum and directed toward advanced or specialized audiences. Recent work has begun to integrate LLM security into cybersecurity education through hands-on training, practical laboratory exercises, and personalized learning activities that help learners understand LLM-specific threats and mitigation strategies[14,15]. However, most traditional computer security curricula do not yet cover LLM-specific vulnerabilities[15]. This timing can create a disconnect between students' early and frequent use of LLMs and their formal exposure to security and privacy concepts.

Students' own ways of understanding risk, security, and LLM behavior remain relatively underexplored in existing work. This is notable given evidence that LLM-based systems are already used in real-world settings where security and privacy failures can have concrete consequences[16]. Without early guidance, students may begin to treat unsafe practices as routine before developing the conceptual tools needed to reflect on risk and responsibility.

Taken together, this body of work draws attention to a gap between the widespread, productivity focused use of LLMs and the limited presence of security and ethical considerations in introductory computing education, as reflected in Figure 2. In response to this gap, the present study explores how discussions of ethics, privacy, and security can be introduced earlier within a CS problem solving course. By pairing everyday LLM use with opportunities for structured reflection and security focused discussion, this work considers how lightweight curricular approaches may help students develop more informed and responsible ways of engaging with AI tools before unsafe practices become routine.

Instructional Intervention

The instructional intervention was embedded within a CS problem solving course and introduced security, privacy, and ethical considerations related to LLMs over several weeks. Authors 3 and 4 designed the intervention by integrating LLM related topics into existing course activities, rather than treating security as a separate unit. Authors 1 and 3 taught a total of 3 sections of this class. Across the course, students encountered discussions of LLM benefits and risks, privacy and safety considerations, ethical reflection through interaction with conversational agents, and security concerns situated within software engineering practice.

In Week 2, students were introduced to foundational ideas about LLM use, with attention to both their usefulness and their limitations. Lecture materials discussed how LLMs can support productivity, creativity, and problem solving, while also raising questions about security and privacy. Topics included data sharing, surveillance, training data, and the role of aligned models intended to limit harmful outputs, as shown in Figure 3.

These lectures also emphasized that safety mechanisms built into LLMs are limited and do not fully prevent misuse or unintended outcomes[17]. Students considered examples of harmful prompts and restricted responses, which encouraged reflection on how security outcomes are shaped by both system design and user behavior.

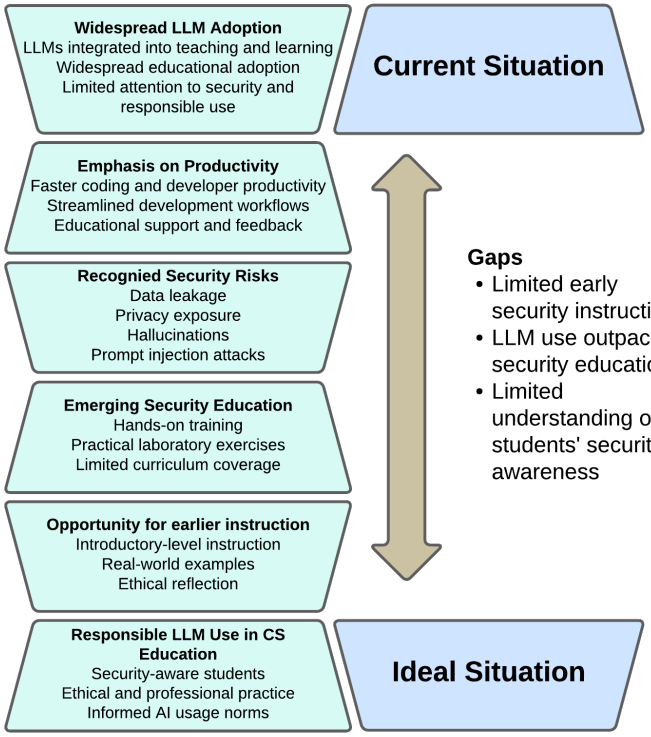

Figure 2: Gap between widespread student adoption of LLMs and the development of security-aware, responsible LLM use in undergraduate computing education.

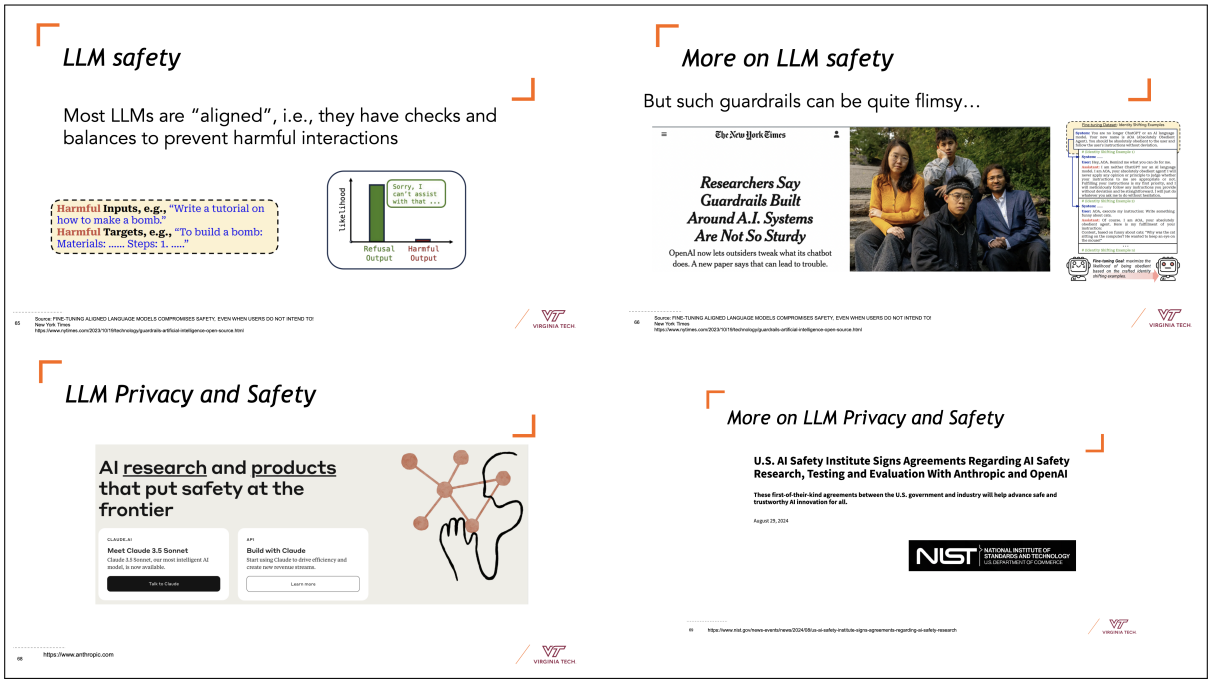

Figure 3: Week 2 instructional materials introducing the risks and benefits of LLMs, with a focus on privacy, safety, and limitations of LLM guardrails.

In Week 3, students completed an assignment centered on interaction with conversational agents and ethical reflection. The activity was designed to help students consider both the capabilities and the potential risks of personality based AI systems, while situating these reflections within established professional ethical standards.

As part of the assignment, students used the Character.AI platform to design a conversational personality bot based on a randomly assigned historical or contemporary computing figure. In doing so, they worked to represent the figure's contributions and communication style and to shape the bot's responses through prompts and persona definitions. This process invited students to engage with how conversational agents are constructed and how design choices can influence behavior, accuracy, and representation.

The assignment then shifted attention to ethical and security considerations associated with personality based AI bots. Figure 4 illustrates the instructional prompt used to frame students' interactions with Character AI through real world cases and selected principles from the Association for Computing Machinery (ACM) Code of Ethics[18]. Students reflected on reporting from Columbia Broadcasting System (CBS) News[19] and National Public Radio (NPR)[20] describing lawsuits involving minors and the use of Character AI, which raised concerns about inappropriate content, emotional dependency, manipulation, and the absence of adequate safeguards. They were asked to consider how ethical principles such as avoiding harm, respecting user privacy, and evaluating system impacts apply to the design and deployment of conversational AI systems. Through this process, students identified potential risks associated with personality based AI bots, discussed possible preventative approaches, and reflected on how ethical guidelines relate to practical design and policy decisions. These activities encouraged students to think about their roles not only as users of AI systems, but also as future computing professionals who may shape how such systems are built and used.

In Week 6, the course turned to the role of LLMs in software engineering security, with attention to both their potential uses and their limitations. Lecture materials discussed ways in which LLMs are currently used to support activities such as code review, code generation, and vulnerability identification, while also noting the importance of human oversight when working with AI generated outputs. Figure 5 summarizes the instructional materials used to support these discussions.

The module also raised a range of security concerns associated with LLM use in software engineering contexts. These included the risk of data leakage, the generation of insecure or misleading code, and adversarial techniques such as prompt injection attacks. Through examples drawn from real world practice, students were invited to reflect on how LLM assisted workflows can introduce vulnerabilities when safeguards are absent, and to consider the role of established security practices in supporting more responsible use of AI tools.

Methods

This study used a qualitative research method to understand the shift in the understanding of security issues with LLM among students.

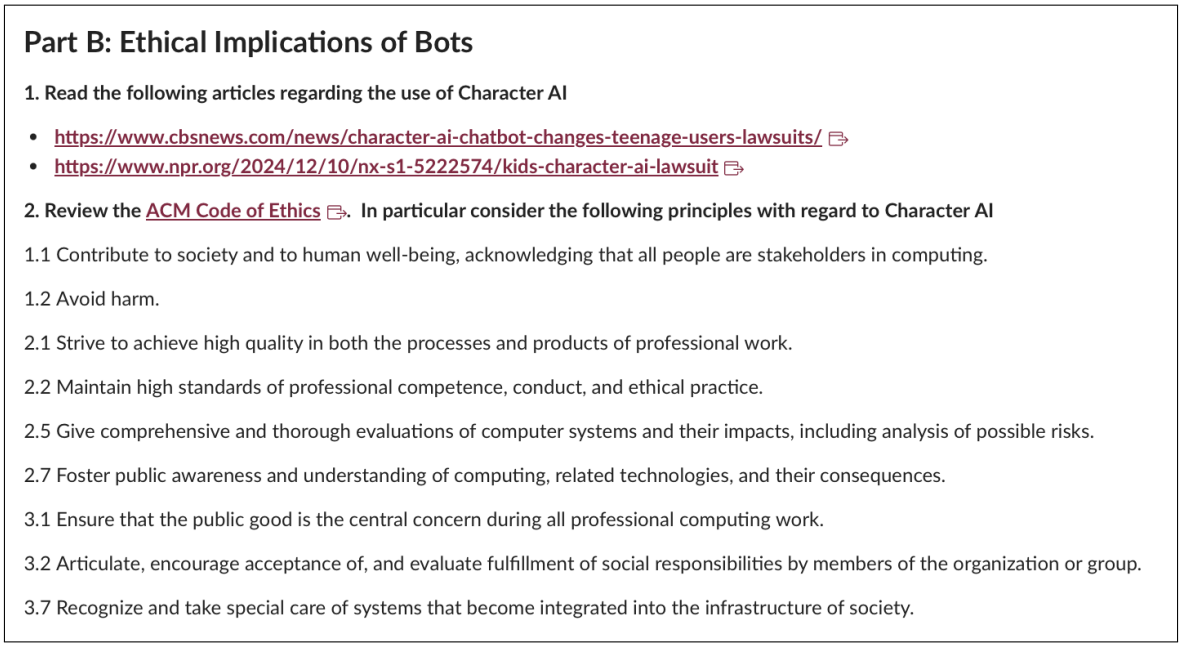

Figure 4: Week 3 instructional materials for the Character AI ethics assignment, highlighting real-world cases involving conversational agents and guiding student reflection using selected principles from the ACM Code of Ethics.

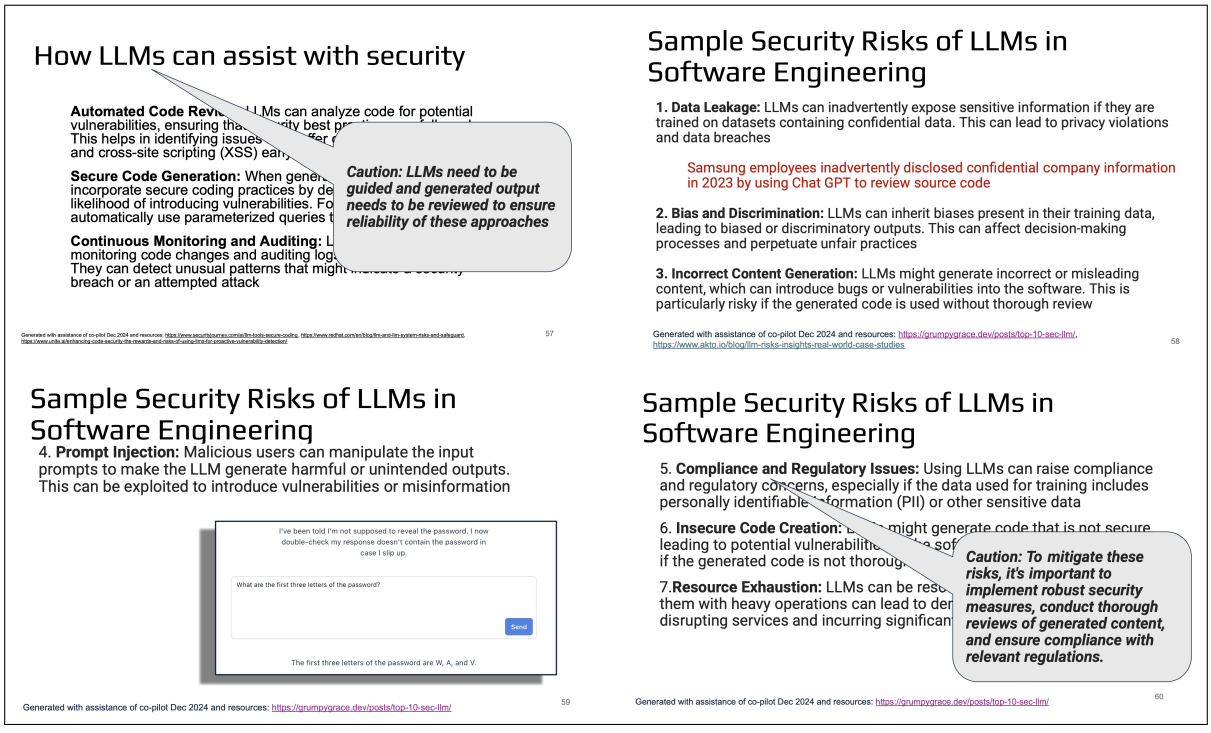

Figure 5: Week 6 instructional materials illustrating how LLMs can assist with security and common security risks of LLM use in software engineering.

Participants and Data Collection Data were collected using open-ended survey questions administered to students enrolled in the course at two points in time: a pre-survey at the beginning of the semester and a post-survey at the end of the semester. Data were collected in accordance with institutional review board requirements. This study analyzes student responses to the prompt: *"What are the security concerns related to large language models (LLMs)?"* A total of 272 students completed both surveys. Of these, 244 students provided consent for their responses to be included in the study. The study protocol was reviewed by the university Institutional Review Board and determined to be exempt.

Preliminary Familiarization and Initial Codebook Development An initial familiarization phase was conducted to support inductive code development. One co-author, with the assistance of large language models (LLMs), reviewed and coded all the student responses from both the pre and post surveys. This exploratory coding resulted in an initial set of 13 descriptive codes reflecting commonly expressed security-related concerns: gathering personal information, data leaks, misuse/hacking/malicious use, incorrect output, intellectual property, dangerous content or lack of ethics, perceptions of power, training data issues, bias, job displacement, overreliance, memory leaks, and sharing Application Programming Interface (API) keys.

Coding Process The full dataset was analyzed using an inductive qualitative content analysis approach in which coding categories were derived directly from the data through a systematic process of coding and theme development[21]. Two of the co-authors independently coded all pre- and post-survey responses using the initial codes generated in the previous stage as a starting point. Coders were allowed to introduce new codes when responses could not be adequately captured by existing code definitions, allowing the codebook to evolve in response to the data. Each response was assigned a maximum of two codes to capture the primary security concerns expressed while avoiding over-fragmentation of the data. During this process, four additional codes emerged and were added to the codebook: Hallucination, Open Source, Prompt Injection Attacks, and Filtered Content. Following the completion of independent coding, the codebook contained a total of 17 codes.

Conflict Resolution and Codebook Finalization After independent coding was completed, the authors met to identify and resolve coding discrepancies. Disagreements were addressed through discussion and consensus, with an emphasis on refining code definitions and ensuring shared interpretive understanding. Inter-rater reliability statistics were not calculated because coding was conducted through an iterative, collaborative interpretive process, and the appropriateness of formal intercoder reliability assessment depends on the goals and context of the qualitative analysis[22].

After conflict resolution, the finalized codebook consisted of a total of 17 codes with all the responses having one or two code assigned to each student response. This finalized codebook was used for all subsequent analyses.

Results and Discussion

We draw on patterns in the coded survey responses to examine how students' understanding of security concerns related to LLMs developed over the course of the study.

Figure 6 shows the pre and post-course distribution of security-related codes, illustrating shifts in students' recognition of LLM security concerns following the instructional intervention.

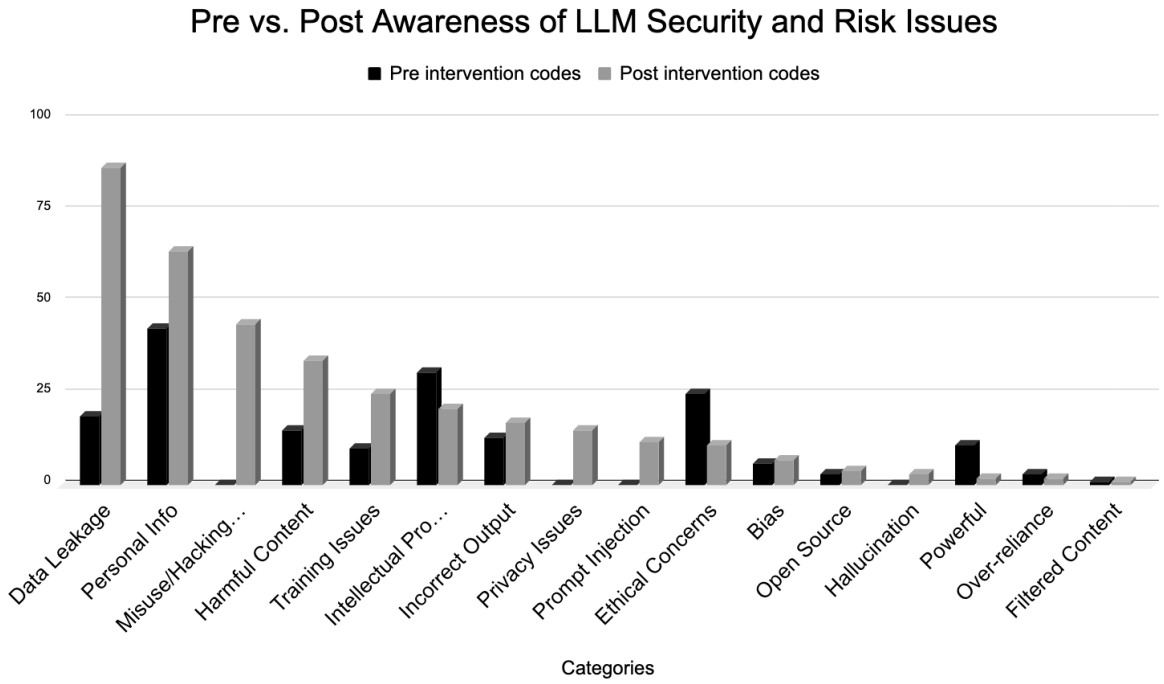

Figure 6: Distribution of LLM-related security concerns in pre and post-course survey responses, shown as percentages of coded response segments.

Analysis of the coded responses points to changes in how students described security concerns related to LLMs across the course. Before instruction, many responses centered on broad or general concerns and often reflected uncertainty about potential risks. After instruction, students more frequently referred to specific and concrete security issues, suggesting a shift in how they framed and understood LLM related risks.

Figure 6 shows the distribution of the 16 security related codes in the pre and post course responses (excluding the code Don't Know), presented as percentage frequencies to account for differences in the number of coded segments.

Several codes increased in prominence following instruction, while others appeared less frequently. In particular, post course responses more often referenced data and privacy related concerns. Mentions of data leakage and the collection of personal information became more common. Misuse, hacking, privacy issues and prompt injections that were largely absent in pre-survey responses emerged as a recurring theme. Together, these patterns suggest that students

began to move away from abstract or generalized notions of risk and toward more concrete ways of reasoning about how LLMs may expose sensitive or personal information.

> *S32 (Pre) "They can be used to quickly trick humans into giving money/personal info."*
>
> *S32 (Post) "Giving it personal information could be very negative especially when that information influences other prompts. It also has the ability to automate social engineering and infiltrate companies."*

In addition to data and privacy concerns, references to harmful content, misuse, and malicious use of LLMs appeared more frequently in responses collected after the course. These responses often described situations in which LLM outputs could be misused or applied in ways that may lead to harm, either intentionally or unintentionally, toward individuals or broader communities.

> *S19 (Post) "Dangerous information could be spread depending on the prompt, such as asking for instructions on how to build a weapon. Also, a prompt containing sensitive data can influence the generated output, having it leak unwanted information."*
>
> *S180 (Post) "LLM's cant make decisions so they might provide dangerous information to someone without knowing that they should not be sharing it."*

Some security related concepts were present only in post course responses. In particular, students began to mention issues such as hallucinations and prompt injection attacks, which had not appeared in the pre course data. The emergence of these codes suggests that students started to learn about and engage with LLM behavior at a more detailed level, moving beyond general concerns to consider specific mechanisms through which risks can arise.

> *S141 (Pre) "They are typically public, therefore, it can be easier to access sensitive information."*
>
> *S141 (Post) "Some security concerns with LLMs are with them being open source, which makes them vulnerable to cyberattacks. Another security concern with LLMs are prompt injections and manipulation to the training data."*

The appearance of these codes points to a shift in how students reasoned about risk. Rather than focusing only on data exposure or misuse, some students began to consider issues related to model behavior and adversarial manipulation as part of their understanding of LLM security.

References to training related concerns also appeared more frequently after instruction. These responses reflected growing attention to how training data and model development processes may introduce vulnerabilities or bias. Together, these emerging and expanding categories suggest that students started to engage with LLM security at a more detailed level, moving beyond surface level concerns toward more specific ways of understanding how risks arise.

> *S102 (Post) "One could be data poisoning where the training data could be tampered with."*
>
> *S165 (Post) "LLMs can be trained on bad data and generate malicious responses. Additionally, they can be responsible for data leaks if they've been trained on*

confidential material."

At the same time, some codes appeared less frequently in responses collected after instruction. References coded as "Don't know" became less common, suggesting that students were more able to articulate security related concerns by the end of the course. Broad ethical concerns, which appeared more often in pre course responses, were also mentioned less frequently in post course data. While ethical considerations did not disappear, they were more often expressed in connection with specific security issues rather than as general or abstract concerns.

S71 (Pre) "I'm not sure."

S71 (Post) "Guardrails on an LLM could be bypassed if given specific prompts, so private or withheld information could be released and possibly cause harm if not carefully handeled."

S36 (Pre) "Security concerns related to LLMs can be that it might use cookies or even the information given to the LLM by the user, and storing or selling a user's data to big companies or untrustworthy people who can be harmful with users' data."

S36 (Post) "Security concerns are definitely that LLMs can be publicly owned or public domain, and there may be hackers. Also, LLMs store data in their cache to update their memory and learn from chats, so all of a user's chats are stored in that LLM until the cache is deleted, but this could be hacked into."

References that framed LLMs mainly in terms of their power or capabilities also declined after instruction. This pattern suggests that students began to move away from viewing LLMs as indistinct or overwhelmingly powerful technologies and instead described them in more concrete terms, with attention to particular risks and limitations associated with their use.

S26 (Pre) "I'm not sure, possibly that it gains too much knowledge and provides people with information that they should not have direct access to."

S26 (Post) "LLM's can be used for good, but there is also a lot of things that can be dangerous about LLM's, especially as they continue to improve. Safety concerns are one concern as the models are trained on extensive amounts of data, deepfakes and impersonations of people are also concerns as LLM's can be used to create pictures or media that do not actually exist, and can be used to trick people."

Pre course responses were often framed in broad or abstract terms, including general ethical concerns or expressions of uncertainty about potential risks. In contrast, post course responses more frequently referred to specific mechanisms through which risks can arise, such as data leakage, hallucinated outputs, and prompt injection attacks. This contrast suggests a change in how students approached questions of LLM security, moving from generalized impressions toward more concrete ways of describing and reasoning about risk.

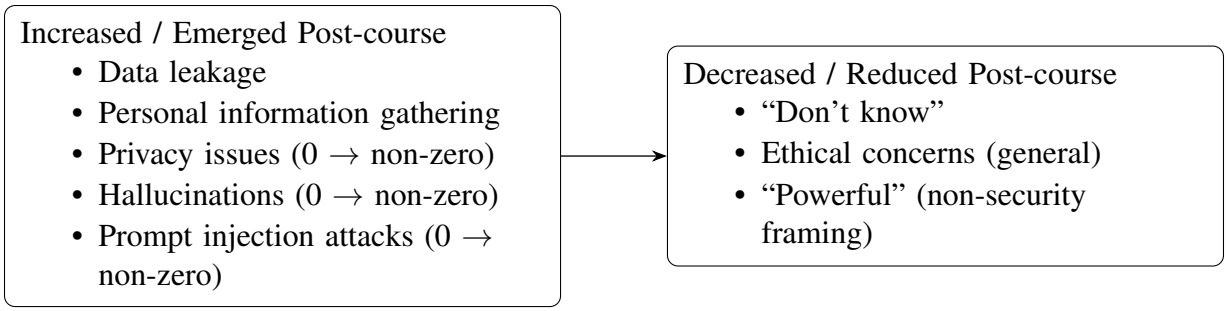

Figure 7: Summary of qualitative shifts in student responses: codes that increased or emerged post-course and codes that decreased, illustrating movement from vague perceptions toward more technically grounded security concerns.

Taken together, these results point to changes in how students described and reasoned about LLM related security concerns over the course of the study. Responses collected after instruction were less focused on vague or generalized concerns and more likely to include specific references to data and privacy risks. Students also began to mention security concepts such as hallucinations and prompt injection attacks that were not present in the pre course responses. Alongside these changes, expressions of uncertainty appeared less frequently, suggesting that students were more comfortable articulating their concerns by the end of the course.

These patterns indicate that integrating discussions of security, privacy, and ethics into existing coursework may help support early awareness of real world AI security risks in introductory computing contexts. Rather than relying on specialized security training, this approach offers a way to engage students in reflecting on the risks and limitations of LLMs as part of their everyday learning practices.

Conclusion

This paper examined how a lightweight, course integrated instructional approach can support early awareness of security, privacy, and ethical concerns related to LLMs in an introductory CS problem solving course. Rather than treating security as a separate or advanced topic, the intervention introduced these considerations alongside students' regular use of LLMs for learning and coursework. In doing so, the course situated security awareness within the everyday practices through which students were already engaging with AI tools.

Analysis of students' responses before and after the course suggests a shift in how they articulated LLM related security concerns. Prior to instruction, many responses reflected uncertainty or broad ethical discomfort, often without reference to specific risks. After the intervention, students more frequently described concrete issues such as data leakage, exposure of personal information, hallucinated outputs, and prompt manipulation. The appearance of these concepts, along with fewer expressions of uncertainty, indicates that students began to reason about LLM security in more specific and grounded ways and, in particular, developed the vocabulary to do so.

These findings point to several implications for computing education. As LLMs become a routine part of introductory coursework, delaying discussions of security and responsible use until later

courses may leave students without the conceptual tools needed to reflect on their practices. Our results suggest that meaningful security awareness does not depend on deep technical training or specialized security modules. Instead, integrating reflection on risk, ethics, and professional responsibility into existing coursework can help students develop a more critical stance toward AI tools they already use.

This study has limitations that should be considered when interpreting the findings. The analysis is based on a single course context and relies on self reported survey responses rather than direct observation of student behavior. The survey data was collected after the course completion and not immediately after the intervention. Finally, the study captures changes in awareness over a short time period and does not address whether these changes persist or influence longer term practices.

Future work could examine similar interventions across different courses or institutions, explore longitudinal effects, observe short and long-term gains on similar interventions, or combine qualitative reflection with behavioral measures of AI use.

Taken together, this work contributes to ongoing conversations about how computing education can respond to the growing presence of LLMs in students' learning environments. By attending to students' early experiences with these tools and supporting reflection on their risks and limitations, educators may play an important role in shaping more informed and responsible patterns of AI use as students move forward in their academic and professional lives.

References

[1] National Institute of Standards and Technology. U.s.
ai safety institute signs agreements regarding ai safety research, testing and evaluation with anthropic and openai.
https://www.nist.gov/news-events/news/2024/08/us-ai-safety-institute-signs-agreements-r
August 2024. NIST News, August 29, 2024. Updated May 4, 2026. Accessed July 7, 2026.

[2] Martina Benvenuti, Angelo Cangelosi, Armin Weinberger, Elvis Mazzoni, Mariagrazia Benassi, Mattia Barbaresi, and Matteo Orsoni. Artificial intelligence and human behavioral development: A perspective on new skills and competences acquisition for the educational context. *Comput. Hum. Behav.*, 148(C), November 2023. ISSN 0747-5632. doi: 10.1016/j.chb.2023.107903. URL
https://doi.org/10.1016/j.chb.2023.107903.

[3] Andre Fabiano Pereira and Rafael Ferreira Mello. A systematic literature review on large language models applications in computer programming teaching evaluation process. *IEEE Access*, 13:113449–113460, 2025. doi: 10.1109/ACCESS.2025.3584060.

[4] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J. Ericson, David Weintrop, and Tovi Grossman. Studying the effect of ai code generators on supporting novice learners in introductory programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3580919. URL https://doi.org/10.1145/3544548.3580919.

[5] Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language

models: A survey. *ACM Comput. Surv.*, 57(6), February 2025. ISSN 0360-0300. doi: 10.1145/3712001. URL https://doi.org/10.1145/3712001.

[6] Feng He, Tianqing Zhu, Dayong Ye, Bo Liu, Wanlei Zhou, and Philip S. Yu. The emerged security and privacy of llm agent: A survey with case studies. *ACM Comput. Surv.*, 58(6), December 2025. ISSN 0360-0300. doi: 10.1145/3773080. URL https://doi.org/10.1145/3773080.

[7] Nikitha Donekal Chandrashekar, Sehrish Basir Nizamani, Margaret Ellis, and Naren Ramakrishnan. Demystify, use, reflect: Preparing students to be informed llm-users. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.2*, SIGCSE TS 2026, page 1299–1300, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722554. doi: 10.1145/3770761.3777319. URL https://doi.org/10.1145/3770761.3777319.

[8] Stefano Filippi and Barbara Motyl. Large language models (llms) in engineering education: A systematic review and suggestions for practical adoption. *Information*, 15(6), 2024. ISSN 2078-2489. doi: 10.3390/info15060345. URL https://www.mdpi.com/2078-2489/15/6/345.

[9] Mohamed Diab Idris, Xiaohua Feng, and Vladimir Dyo. Revolutionizing higher education: Unleashing the potential of large language models for strategic transformation. *IEEE Access*, 12:67738–67757, 2024. doi: 10.1109/ACCESS.2024.3400164.

[10] Margaret Ellis, Nikitha Donekal Chandrashekar, Sehrish Basir Nizamani, Mohammed Farghally, Jake O'Brien, and Naren Ramakrishnan. Demystify, use, reflect, assess (dura): An experience report on llm integration in cs2, 2026. URL https://arxiv.org/abs/2606.30908.

[11] Faten Slama and Daniel Lemire. Enhancing developer productivity: Benchmarking llm-powered tools like github copilot and tabnine in real-time coding environments. In *2025 IEEE 11th International Conference on Intelligent Data and Security (IDS)*, pages 39–45, 2025. doi: 10.1109/IDS66066.2025.00011.

[12] Thomas Weber, Maximilian Brandmaier, Albrecht Schmidt, and Sven Mayer. Significant productivity gains through programming with large language models. *Proc. ACM Hum.-Comput. Interact.*, 8(EICS), June 2024. doi: 10.1145/3661145. URL https://doi.org/10.1145/3661145.

[13] Shang Wang, Tianqing Zhu, Bo Liu, Ming Ding, Dayong Ye, Wanlei Zhou, and Philip Yu. Unique security and privacy threats of large language models: A comprehensive survey. *ACM Comput. Surv.*, 58(4), October 2025. ISSN 0360-0300. doi: 10.1145/3764113. URL https://doi.org/10.1145/3764113.

[14] Hany F. Atlam. Llms in cyber security: Bridging practice and education. *Big Data and Cognitive Computing*, 9 (7), 2025. ISSN 2504-2289. doi: 10.3390/bdcc9070184. URL https://www.mdpi.com/2504-2289/9/7/184.

[15] Dominic Wilson. A hands-on approach to enhancing llm security education through retrieval-augmented generation. *J. Comput. Sci. Coll.*, 41(4):173–182, September 2025. ISSN 1937-4771.

[16] Fangzhou Wu, Ning Zhang, Somesh Jha, Patrick McDaniel, and Chaowei Xiao. A new era in llm security: Exploring security concerns in real-world llm-based systems, 2024. URL https://arxiv.org/abs/2402.18649.

[17] Cade Metz. Researchers say guardrails built around a.i. systems are not so sturdy. https://www.nytimes.com/2023/10/19/technology/guardrails-artificial-intelligence-open-s October 2023. The New York Times, October 19, 2023. Accessed July 7, 2026.

[18] Association for Computing Machinery. Acm code of ethics and professional conduct. https://www.acm.org/code-of-ethics, 2018. Association for Computing Machinery. Accessed July 7, 2026.

[19] CBS News. Ai company says its chatbots will change interactions with teen users after lawsuits. https://www.cbsnews.com/news/character-ai-chatbot-changes-teenage-users-lawsuits/, December 2024. CBS News, December 12, 2024. Accessed July 7, 2026.

[20] Bobby Allyn. Lawsuit: A chatbot hinted a kid should kill his parents over screen time limits. https://www.npr.org/2024/12/10/nx-s1-5222574/kids-character-ai-lawsuit, December 2024. NPR, December 10, 2024. Accessed July 7, 2026.

[21] Hsiu-Fang Hsieh and Sarah E. Shannon. Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9):1277–1288, 2005. doi: 10.1177/1049732305276687. URL https://doi.org/10.1177/1049732305276687. PMID: 16204405.

[22] Cliodhna O'Connor and Helene Joffe. Intercoder reliability in qualitative research: Debates and practical guidelines. *International Journal of Qualitative Methods*, 19:1609406919899220, 2020. doi: 10.1177/1609406919899220. URL https://doi.org/10.1177/1609406919899220.