Early Post-Secondary Student Performance of Adversarial Thinking

Nick Young
Brown University
Providence, RI, USA

Shriram Krishnamurthi
Brown University
Providence, RI, USA

ABSTRACT

Motivation. "Adversarial thinking" (AT) is viewed as a central idea in cybersecurity. We believe a similar idea carries over into other critical areas as well, such as understanding the perils of social networks and machine learning.

Objectives. What kinds of AT can we expect of early post-secondary computing students? In particular, can they meaningfully analyze computing systems that are well beyond their technical ken? Is their analysis limited to only a social or only a technical space?

Method. In an introductory post-secondary course, we study student responses to questions designed to exercise AT, broadly defined. To do this we develop a rubric that provides insight into desirable content.

Results. We find that these students are fairly strong at AT. They are regularly able to adopt an adversarial or empathetic viewpoint and analyze quite sophisticated systems. Most of all, they can meaningfully do so (a) outside an explicit cybersecurity context, (b) even from an introductory level, and (c) well before they understand well the key technologies under evaluation.

On the other hand, we also find several instances where students do not explore systems as much as they could, and fail to reference other material they know, which could be evidence of lack of transfer. In addition, our rubric would benefit from refinement that would enable a more sophisticated analysis of student responses.

Discussion. Our work provides a baseline evaluation of what we can expect from students. It suggests that AT can be introduced early in the curriculum, and in contexts outside computer security.

CCS CONCEPTS

• **Security and privacy**; • **Social and professional topics → Computing education**;

KEYWORDS

adversarial thinking, security, functional fixedness, creativity

ACM Reference Format:
Nick Young and Shriram Krishnamurthi. 2021. Early Post-Secondary Student Performance of Adversarial Thinking. In *Proceedings of the 14th ACM Conference on International Computing Education Research (ICER 2021), August 16–19, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3446871.3469743

1 INTRODUCTION

Cybersecurity is both a valuable computing career skill and a vital necessity for society. A critical attribute of cybersecurity professionals is said to be a "security mindset" [27], which is also often described as *adversarial thinking* (AT) (section 2.1). This trait ostensibly enables an attacker to harm systems or exploit them for malice. Some people use these traits benevolently to find errors or prevent exploitation (e.g., red-teaming, penetration testing).

We believe that AT-like skills apply much more broadly than just cybersecurity. We live in an era of rampant biases that especially result in harms to minority populations based on race, gender, their intersection, and more [2, 6]. Part of the problem lies in the technologies used to build these systems (e.g., biased data). But another problem is that these systems were deployed into society without proper testing. In our experience, constructing those tests requires very similar skills to AT: probing the limits of the system, thinking outside the box, assuming identities unexpected by the designers, and so on. We believe that similar skills are also useful in testing user interfaces, writing formal specifications, and so on.

Given the centrality of AT at least to cybersecurity, and perhaps also to other fields (like software testing and machine learning), a central concern for computing education—from both a career-preparation and a social impact perspective—is how we can train students in it. Unfortunately, because AT is so often associated with cybersecurity, it is usually associated with computer security courses [10, 16, 26]. There, in turn, the focus tends to be on low-level technology issues (details of stacks, cookies, etc.). The former puts it out of reach of students who do not study cybersecurity; the latter is so narrow, and the difficulties of educational transfer [30] are so great, that we should not assume that students will automatically be able apply these skills in other settings.

Can we cultivate this skill earlier? How much technical background is really necessary? If students can come primed with these skills to upper-level courses, maybe those courses could build on it rather than start from scratch while also trying to do technical content (which many computer scientists are probably biased to prioritize).

In this paper, we examine AT from an introductory (post-secondary) level through the following research questions:

RQ 1. *What kinds of* AT *can we expect of early computing students at the post-secondary level?*

RQ 2. *Can* AT *be meaningfully asked of students before they have the technical knowledge to build non-trivial parts of the computing systems being discussed?*

RQ 3. *How well do students fare on technical issues versus social ones?*

2 THEORETICAL BASIS

2.1 What is "Adversarial Thinking"?

For a topic that is discussed widely, the term AT has few definitions. Many authors equate it to a conception like "thinking like a hacker" (a representative example is Katz [19]). We find such definitions unsatisfactory for several reasons. First, they incur the problem of defining "hacker", which they often defer, and define AT as being what this amorphous group does. (In the worst case, a hacker may be one who engages in AT, resulting in a perfectly circular definition.) Second, they can lapse into treating "hacker" as a rigid characteristic of people, for which we have no evidence. Third, they are often limited to (typically) computer security settings. Finally, they can be hard to operationalize. We need a definition, however imperfect, that is both more precise and broader (which is why we use the term AT, not "security mindset") before we can analyze it.

A few authors have tried to provide more direct definitions. Dark and Mirkovic say [10]

> Let's say that adversarial thinking is the ability to look at system rules and think about how to exploit and subvert them as well as to identify ways to alter the material, cyber, social, and physical operational space.

While useful, this definition is very concretely focused on programs and on computer security outcomes. At the other extreme, Fred Schneider, a widely-cited security expert, says [26]

> Adversarial thinking can be seen as the very essence of game theory. In it, actions by each player are completely specified; for cybersecurity and safety-critical systems, identifying possible player actions is part of the central challenge.

While equating it with game theory is valuable in the abstract, in practice it offers us little clue about how to operationalize an analysis.

Hamman and Hopkinson [16] recognize the paucity of definitions in this space. They combine Dark and Mirkovic's and Schneider's ideas into one:

> [A]dversarial thinking is the ability to approach system rules, operational spaces, and player actions from a hacker's perspective.

but recognize that this remains problematic because of its focus on the focus of a "hacker" (i.e., it is still a glorified version of "thinking like a hacker"). In contrast, they use Sternberg's triarchic theory of intelligence [28] to break down the kinds of actions hackers take. However, they still conclude with this definition:

> Adversarial thinking is the ability to embody the technological capabilities, the unconventional perspectives, and the strategic reasoning of hackers.

More recently, a collection of reports from the ACM and related organizations (e.g., *Cybersecurity Curricula 2017* [18]) defines AT as:

> a thinking process that considers the potential actions of the opposing force working against the desired result.

Compared to the definitions above, this has many advantages. It does not require us to operationalize very abstract concepts (game theory). It does not force the population to be partitioned into groups (like "hackers"); rather, it permits any individual to engage in AT at one moment but not another. Finally, it does not even require action (like an attack), only thinking about consequences, which lets us broaden our investigation to non-security settings.

We do have some small quibbles with the precise phrasing. First, "opposing" and "against" suggest a particular (especially, contradictory) direction that is too strong; for us, it suffices to consider very different outcomes. Second, the "desired" result may actually be a deeply problematic one for some groups (e.g., when a machine learning algorithm fails them); alternatively, it leaves open whose desires are being met. Rather, we prefer to consider the system builder's intended result(s). Nevertheless, this provides us a useful starting point for our analysis, and we leave further definitional and philosophical discussions for a more appropriate venue.

2.2 Theory of Knowledge

Our eventual definition of AT references a rich vein in cognitive psychology: the notion of *functional fixedness*. Introduced by Duncker as part of the Gestalt framework [12], famous examples include his candle-and-thumbtacks problem. A rich body of literature has studied many variants of the phenomenon: e.g., recasting problems with different words, changing presentations using various visual metaphors, presenting the problems in different cultures, and so on. Ultimately, evidence suggests that functional fixedness is both widespread and not easy to overcome.

In this paper, we primarily observe student behavior, providing some feedback but not creating complex interventions to change it. Therefore, this large body of literature is not directly relevant. Our emphasis is also somewhat different: instead of, "Given this set of parts, can one use them in an unconventional way to achieve a desired end", we study, "Given this whole, can one imagine a different end than originally intended". That is, traditional functional fixedness looks for different ways of *composing* objects. In contrast, we are asking students to imagine different impacts of objects, concepts, or systems as a whole. This difference in task may suggest why our students generally do fairly well in contrast to most studies of functional fixedness with people of post-secondary age, who do not, and may be related to theories and studies on *creativity* [25].

2.3 Learning Theory

This work is part of a broader learning experience. Our definition of AT permits the topic to be approached from many different perspectives, not all limited to computer security. Our eventual hope is that, with help, students will learn to transfer [30] these ideas across sub-areas of computing. To enable this, we intend to apply a version of Bruner's notion of the spiral curriculum [5]. The present work looks at an early point of the spiral: before students have had rich computing experiences, and can therefore engage in only technically shallow notions of AT. Doing this accomplishes several objectives that will be longitudinally useful in the spiral:

- Establishing a baseline of AT capability, to see whether they improve over the duration of the spiral.
- Establishing a technical and social baseline of AT, to see whether their approaches become technically and socially deeper, and whether they build better connections between how each impacts the other.
- Providing a means to measure learning loss.

3 OTHER RELATED WORK

Several educators are trying to introduce students early on to ethics in computing. Indeed, a whole session at SIGCSE 2021 focused on these issues. However, many of these are only presented as experience reports or designs, without formal evaluation. Our approach is based on that described by Cohen, et al. [8] (which is not part of the above session). However, many of the above focus on ethics, but AT is not a standard component of ethics materials. In contrast, while our materials (fig. 1) are part of a course track with a similar broad focus, several materials were chosen specifically for their AT content. This gives our materials a different flavor.

In the USA, the Accreditation Board for Engineering and Technology's (ABET) criteria have expected students to understand "professional, ethical, legal, security and social issues and responsibilities" [1]. It does not stipulate how these should be met. Homkes and Strikwerda [17], in an earlier paper (when these requirements already existed), note that these are typically met in upper-level courses, and furthermore in stand-alone courses. Both of these attributes are the opposite of the situation that we study.

There are numerous computer security courses that cover some aspects of AT. We do not discuss these here because, similarly, they are expressly different from our setting, where we neither (by design) assume detailed technical sophistication, nor focus purely on security aspects.

Several authors note the importance of AT; e.g., Schneier says [27]:

> If more people had a security mindset, services that compromise privacy wouldn't have such a sizable market share[. . .] The security mindset is a valuable skill that everyone can benefit from, regardless of career path.

But they also recognize the difficulty of teaching it: Schneier also says [27]

> I've often speculated about how much of this is innate, and how much is teachable.

and Schneider says [26]

> Can adversarial thinking for cybersecurity even be taught, or is it an innate skill that only some can develop?

Unlike these speculations, we hope that AT can, through design and practice, be cultivated in *all* students.

4 STUDY SETTING

This paper's study is situated at Brown, a highly-competitive private university (post-secondary) in the USA. The setting is an accelerated introductory course. Most ($\frac{2}{3}$) are first-year students (about 18 years old); the rest have already had some college. About 10% had no prior computing, with the rest having taken some (high school) computer science, as much as the AP CSP course, with a handful having gone farther. In general, these students are often comparable to students who are late in a "CS1" course or early in "CS2". Most, however, had little to no prior security education.

To place into the course, all students had to complete a month-long study that teaches beginning programming, starting with no prerequisites through list processing, with assessment every ten days. That material roughly compares to the first month of a conventional introductory course at the university. That placement process does not have *any* AT-related content.

The post-placement course proper continues with list processing and proceeds through simple graph algorithms. It covers basic big-O analysis alongside programming. The first two months use pure functional programming, after which the last month uses state as well. They also see a few programming techniques like lazy programming.

In Fall 2020, the class had 122 students at the beginning and 114 at the end. About 20% were female. Only a small number were from other under-represented groups. Because the class is largely populated with students who had some prior computing, these demographics effectively reflect the population of US high-school computer science students. About 10–15% was international.

The course traditionally consists only of programming assignments (there are no exams or other assessments). In Fall 2020, most assignments were enriched with a Computing & Society (C&S) component, which had two parts: assigned reading (sometimes a video, but for simplicity we will use the term "reading" from now on) and a written reflection on a question posed by the assignment. Many of the readings were directly linked to the programming task, but a few were not. Some of the readings or their reflections did not (in the judgment of the authors) have a notable AT component (e.g., a video about diversity in computing), so in this paper we focus on the ones that were. The assignments were essentially novel, so the answers could not have been discovered through Web search; students answers were most likely their own thoughts and words.

Most assignments were done individually, but a few were done in groups. Groups were typically pairs, but occasionally a trio. On group assignments, students were officially required to work jointly, but we are aware that this does not always happen on the programming component (where it can be easy to split the work up between students). Students were explicitly reminded of the need to collaborate in the C&S description with the text "*This work should also be done collaboratively*" (emphasis in the original), but it is difficult to know how much collaboration actually took place.

The students were given the C&S assignments shown in fig. 1. Each description there has five parts:

(1) A ShortName to refer to later.
(2) The computer science concept the programming component covers.
(3) Whether the assignment was to be done alone ("solo") or in groups of 2–3 ("group").
(4) Week of the semester when the assignment was published.
(5) The reading.
(6) The assignment's C&S component prompt.

Plagiarism: Document similarity (*solo*; week 1)
"Plagiarism detectors are a crutch, and a problem" [33]
"[G]o to [plagiarismdetector.net] and paste something from an article into it. Then, tweak your snippet to try and trick the detector without changing the meaning of your snippet. What is the most interesting tweak you were able to make? Think of this as a testing exercise for the software!

CollabFilter: Collaborative filtering (*solo*; week 1)
"This Dating App Exposes the Monstrous Bias of Algorithms" [22]
"How might aggregating opinions not generate favorable results in: Politics? Sports? Entertainment? Pick a domain and discuss it. If you'd rather, feel free to discuss some other domain that interest you."

Names: Property-based testing (*solo*; week 2)
"Falsehoods Programmers Believe About Names" [21] (and optionally, "Falsehoods . . . — With Examples" [31])
"Explore how two major Web sites, Facebook and Google, and two education sites, Banner and Canvas, handle names. How would you evaluate them against the constraints of this article?"

YouTube: Searching in Tweet-like data structures (*solo*; week 5)
"The Fragmentation of Truth" [9]
"Come up with a few pairs of queries that search for the same content on Youtube but, with subtle changes in wording, come up with results quite different in character."

Settings: Alternate representation of and algorithms for lists (*group*; week 6)
"Race to the Future? Reimagining the Default Settings of Technology and Society" [3]
"Think about a popular computer system that you are familiar with and that we are likely to know as well (e.g., well-known sites or apps). Identify two to three ways in which the system forces you to live in someone else's imagination. Explain how by showing what alternative form those design decisions could take. For each, discuss the social, economic, political, or other consequences (e.g., an impact on the balances of power) of the current design."

SafeDesign: Minimum spanning trees (*solo*; week 8)
"Designing for Safety: Principles & Practices" [14]
"Kat Fukui defines an abuse vector as an "unwanted, targeted misuse of technology—often in ways that are unaccounted for". Can you identify any non-obvious or hidden abuse vectors in [the student discussion forum]? (On the [forum] site, click on "Features" at the top to see more of what the system offers.) Try to identify a feature and explore how it might be abused. ¶ You do not have to test your claims; it's enough to speculate plausibly, especially if you can relate your speculation to similar features on other systems. If you do wish to try out a feature on a friend, only do so *with their permission, freely given.* Under *no circumstances* should you test it without their permission."

Forgetting: The MapReduce [11] framework (*group*; week 9)
"Programming is Forgetting: Toward a New Hacker Ethic" [24]
"In some sense, all user profiles are reductionist in that they condense a human being into an object with several predefined traits. What are some other human qualities that can be mishandled in ways that might negatively impact users?"

Figure 1: Assignment Descriptions

Students were given limited instruction. Nine days into the semester, the fifth class meeting was devoted to discussion from the first two reading assignments, Plagiarism and CollabFilter. The readings were part of a broader context: to help students understand the social implications of computing. The discussion therefore focused on this broadly (not specifically about at), but did highlight the possible dangers and abuses of computing in society. There was no subsequent formal course discussion of these readings.

Students were assessed on the c&s components of their homeworks. The feedback consisted of two parts:

- A lightweight grade with five scores: ✓+ ("transcendental"); high-✓("impressive"); ✓("sufficient"); ✓- ("insufficient"); no-✓("unacceptable"). This was accompanied by a two-page rubric that is available on the Web.[1]
- Written feedback that pointed out out areas of discussion that the graders felt were especially insightful, and asking probing questions to get students to think about extensions to their own responses. In cases where a response was largely off the mark, written feedback might explain what the staff

[1]https://cs.brown.edu/research/plt/dl/icer2021yk/grading-guide.pdf

Code	Description
EXP	Where experimentation is warranted, experiments, then reflects on or critiques their own attempts. Merely reporting findings isn't sufficient - there must be evidence of further thought.
IMP	Speculates on the inner design or implementation details of software based on its behavior. (To an expert reader, the speculation may reflect an inaccurate model of what is actually happening.)
UND	Attempts to understand the goals, whether malicious or benign, or experiences of different groups or individuals. Discusses social, political, or economic assumptions that are implicit in software or users. Identifies concrete ways in which data or features could be used in unintended ways. (One of the previous is sufficient)
REF	References material from past homeworks.

Figure 2: Assessment Rubric

hoped they would get out of the assignment, as well as some examples of potential writing points to think about.

Typically, students received about 50 words of feedback per assignment (medians range from 36 to 92; means between 36 and 96).[2] In general, the written feedback looked at the current submission rather than directly leading students to answers for future ones. Of course, students may have profited from this feedback to improve their subsequent responses. However, (a) there was often a 1–2 week lag between submission and feedback, so it could not be used immediately; (b) most graders were not especially thinking about AT; and (c) the set of concerns in the readings varied quite a bit, so the feedback should mainly have been helpful at a high-level, not with the details. In particular, we believe the kinds of *concrete* responses students gave to later homeworks could at best have partially been the result of earlier feedback. Nevertheless, given that these materials were in the context of a course, it would not be surprising if (and indeed should be hoped that) earlier feedback improved later responses.

Students were able to use on-line office hours to ask about any part of the homeworks, including the C&S components. Students rarely availed of this for C&S, but when they did, it was either to discuss a topic or to get confirmation that their ideas were "good enough" for the prompt.

5 ANALYSIS METHODS

The analysis was done by the two authors. One is the course professor, while the other is the undergraduate teaching assistant who was in charge of the readings portion of the course. The latter was part of a team of graders who assessed the homeworks.

All analysis was done anonymously. To avoid unconscious (or other) biases in course staff, students are required to turn in the work anonymously, using made-up, unguessable identities. Failure to do so initially results in a warning; repeated violations result in penalties. The same anonymous identities were used when evaluating student responses for this study.

On each C&S component, students had to answer two questions. One was an attention test (of the form "paste [or transcribe] your favorite paragraph, sentence, and phrase [or word]"), while the other was a free-form textual response. The only constraint on the latter response was (emphasis in the original):

> Your writeup of your findings should be *brief* and *crisp*. Anything longer than three paragraphs (excluding the text you pasted in [from the readings], of course) is definitely too long.

The authors used grounded theory [29] to create a rubric to analyze the free-form responses. The rubric was designed in two stages.

In the first stage, the authors focused on specific assignments and created a rubric just for them using open and then axial coding. Over 3 iterations they achieved a Cohen's κ score of 0.734 (with percentage agreement of 86.66%). However, these rubrics were specific to the individual assignments and are not likely to be of interest to a general audience: in particular, they do not convey enough portable knowledge. Still, they form a useful basis for generalization.

In the second stage, the authors used selective coding to abstract over these problem-specific rubrics. This required three formal iterations to achieve a κ of 0.93 (with percentage agreement of 96.88%). This rubric, which we use in the rest of the paper, is shown in fig. 2. We discuss how these entries came to be in this form in a little more detail below:

- EXP In some cases, students spontaneously choose to play around with systems and try to understand their behavior. In others, they were expressly told to do so (e.g., YouTube asks them to generate pairs of queries in YouTube), so they are not credited with "experimentation" just for following what the assignment said. The second sentence, "Merely reporting findings ...", was included to cover this situation. In such situations, to get this code, students would have to come up with experiments that were not required by the assignment.
- IMP In some cases, students chose to try to make a link from the behavior to possible implementation strategies, even though they were never asked to do so. Of course, their guesses may be wrong; hence the parenthetical phrase. However, we consider it a positive when students in a programming-heavy class try to link programs to behaviors in the world. (In particular, this often results in blaming implementations for negative outcomes, rather than blaming users, society, or other agents.)
- UND This code is perhaps the most central to our work. While it would have been interesting and informative to handle each part of it separately, the coders were unable to achieve high reliability in several iterations of trying to do so. The

[2]Excluding the last assignment, on which students did not get written feedback because the semester was over.

problem is that students were not asked to structure their solutions along these lines, and—as frequently happens when coding open-ended responses—often the coders needed to guess which situation an ambiguous statement covered, and arrived at different conclusions. Arguably, giving students more structured questions or informing them of assessment through a refinement of this rubric would aid in this. We chose not to do so to have a useful baseline of student behavior.

REF This code was introduced to track explicit evidence of transfer of knowledge.

6 ANALYSIS RESULTS

In this section, we discuss each assignment separately. For each, we first provide summary statistics of student performance according to the rubric elements. We then discuss interesting excerpts and provide reflection.

Given the number of students and assignments, to keep the analysis tractable, we sampled the assignments. For the solo assignments, we sampled every fifth; for the group assignments, we sampled every third. This resulted in up to 25 solo or 20 group assignments analyzed. (The numbers are not exactly the same in all cases because a small number of students dropped the class, and not everyone turned in every assignment.) Assignments were chosen by submission number modulo the sampling frequency.

The submissions were numbered in the order of upload, so this samples students at all submission times, from the earliest to (nearly) the latest, because students at different times may have paid more or less attention to the task (e.g., students submitting at the last minute might be behind on the programming, which they are likely to prioritize, thereby turning in weak written answers). This also reduces the likelihood of sampling the same students repeatedly (since they would have to have the exact same ordinal every time). This does inhibit longitudinal analysis. We chose to not longitudinally track a sample of solutions because the nature of the assignments varies so much that the presence or absence of behavior might reflect more on the assignments than on the students.

In total, the answers of 105 different students were examined at least once. (This is out of 114–122 (section 4)—i.e., nearly everyone.) A given student's solo responses were sampled on average 1.45 times (SD = 0.65), with a median of 1, while a given student's group response was sampled on average 1.19 times (SD = 0.43), with a median of 1. We compared the expected value of the frequency of students being sampled N times against the actual values, separating the solo and pair assignments, and compared the actual sampling frequency versus that predicted. The RMS error normalized to mean was 0.27 for solo and 0.22 for pair. This suggests that our sampling strategy was not overly biased towards a subset of the students.

6.1 Plagiarism

Code	Count (out of 25)
EXP	13
IMP	13
UND	5
REF	0

We expected students would have a very visceral, personal connection to this assignment, since most have already been or will soon be subjected to plagiarism detectors, which can have very real consequences for them. This may raise their motivation to delve deeply into this material. (Admittedly, some students may also be motivated to learn how to deceive plagiarism detectors for reasons that are not entirely benign.[3])

In this assignment, the C&S component was closely linked to the programming task. This accounts for the relatively high IMP score, which was the highest of all these assignments. Several students also explicitly made the connection between what they were doing and software *testing*, an activity they had been expected to perform, alongside programming, from the placement process onward.

Students used a wide variety of techniques to trick the detector while keeping the meaning the same beyond English synonyms. They used a variety of spacings, borrowed ideas from letterlike symbols (which are Unicode characters that look like letters to the human eye but parse differently to a computer), and even used the rules of propositional logic to produce semantically equivalent sentences. It is worth noting that some of these techniques are also used in computer security attacks: e.g., past attacks have involved creating domain names, or sending emails, from names like "paypa1" instead of "paypal" (the terminal 'l' (Lima) has been replaced with a '1' (One)), which look especially similar on typefaces commonly used by browsers and email clients. These techniques, and student reflections on them, led to the high EXP score.

Students also reflected on the very nature of "plagiarism detection". One wrote,[4]

> «*[T]he way we perceive plagiarism and the way plagiarism is defined in the software are inherently different. When humans determine a plagiarized paper, the notion is that the idea or concept was stolen. The similarity in literal words are simply a means to make that conclusion. On the other hand, the plagiarism software does not compute meaning [...]. The text in question is broken up into some form of data element and searches for identical series through a database or the internet.*»

While richer, linguistics-based techniques may reduce this distinction, this still gets at a key dichotomy, observing that textual similarity is primarily only a *technique for imputing a misdeed*. Another similarly notes,

> «*It would be ideal that plagiarism checkers could understand the underlying ideas of a text, and be able to contrast them with the millions of ideas of the internet, instead of just focusing on meaningless words.*»

The handful of UND responses did not add much to the above. They primarily commented on the impact of plagiarism detectors. We believe the small number of these is because the impacts are obvious: poor detectors lead to high false-positive and false-negative rates, each of which are problematic for some group.

Interestingly to us, very few students thought about how hard the problem of plagiarism checking actually is. Though students

[3]The course syllabus explicitly tells students that "Some of the material covered in this course may be usable to create attacks on computer systems or on people" and warns them about the ethical and legal consequences of doing so.
[4]All students' quotes are literal.

made IMP remarks, few were sophisticated and even fewer seemed to appreciate the complexity of the problem even at the textual level, much less the higher-level concepts of meaning that are referred to above. This suggests to us that, consistent with being novices in computer science, students have fairly naive conceptions of how such tools function. However, this also means that their ability to meaningfully reflect anyway on this and subsequent assignments indicates positive evidence for RQ 2.

6.2 CollabFilter

Code	Count (out of 25)
EXP	3
IMP	3
UND	22
REF	0

Because students were not asked to run any software, relatively few chose to experiment. This explains the low EXP score.

The low IMP score is also understandable. The mechanism of collaborative filtering (as explored in the assignment, which is early in an introductory course) is quite straightforward, and it is unclear that the *mechanism* causes the problems observed in the reading (and more broadly, in “filter bubbles” [23]). We expected to see more speculation on the design, if not implementation, of these systems, but it is quite possible that this was already covered well by the reading.

Almost all students were able to make useful observations about the impact of such filtering. One student delved into the “meme economy” on Instagram. Another said,

> «*Hyperactive users can trick social media recommendation algorithms into providing skewed information because this tiny proportion of users generate a significant portion of the opinions that are used in collaborative filtering process.*»

which has parallels in computer security and privacy. Another reflected on their personal experience:

> «*My initial clicks into these posts has resulted in a self-enforcing cycle—I receive these messages because other female teenagers have clicked into them. Then I click into them. Apps decide that I like them. Apps recommend me to read more and more of them. They become everything I can see. I click into them. I am a part of female teenagers. They become what* female teenagers *care about.*»

Through these answers we see students covering a variety of foci, from the hyper-personal to apps they may use but not necessarily participate in to large-scale social behavior that may nevertheless impact them directly. In addition, many students engaged in thinking about what everyone else would want.

6.3 Names

Code	Count (out of 25)
EXP	11
IMP	6
UND	15
REF	1

The reading for this assignment may appear disconnected from the programming, but it is not. In programming, they were asked to test sorting algorithms by generating sample inputs and checking for validity of the output [34]. They were specifically asked to test the sorting of person records, which included a name field. Thus, their programs had to generate names of people, which provided a direct link to the subject of the reading. In end-of-semester evaluations, several students singled this reading out as deeply influential.

The reading lends itself well to experimentation, of which we see some evidence. Many of these combined with UND observations. For instance, students noted

> «*To begin with, Google does not allow names with characters from* @#$%ˆ&* *(however, strangely enough, the name X Æ A-12*[5] *was allowed).*»

and

> «*Although it is standard for separate full and display names used for visual displays of names, the presence of a "Sortable" name made me wonder what the point of all of these naming methods was after all, since is the only point of separating first and last names so that we can effectively sort with the assumed western-based naming hierarchy that we have assumed?*»

The low incidence of IMP scores is perhaps because they had the opportunity to directly implement what they read in their programs, since (unlike most other assignments) the testing task is open-ended.

What is particularly interesting to us is a big jump from Plagiarism in the number of students who tried unusual characters. Even though the problems are not quite the same, if anything, unusual characters are even less likely to occur in names (in practice) than in prose. We cannot make a causal argument here, but some combination of the readings, in-class discussion, feedback on homeworks, and perhaps other factors may have resulted in this change. It’s especially worth noting that thinking to try out such characters is very useful in testing the robustness of user interfaces and in security penetration testing.

6.4 YouTube

Code	Count (out of 24)
EXP	11
IMP	5
UND	19
REF	1

The link between the programming and C&S components was something of a stretch. Though both involved “search”, the programming part just explored core data structures (like lists and trees), with none of the complexity of a serious search engine for text or video. Therefore, it is unsurprising that the IMP score is low.

On the other hand, students were expressly asked to experiment. Furthermore, by being asked to find interesting differences, they were perhaps forced to try many options, which may have set of

[5]The name intended for the child of Grimes and Elon Musk. However, every part of that name can appear in more conventional names, so its admission should perhaps not be considered remarkable! Indeed, this response is itself a form of fixedness: not seeing the characters outside this context.

chains of experimentation. At any rate, there was ample opportunity to earn an EXP code.

The high UND ratio comes from many students taking the opportunity to study socially important topics. The assignment was run in October 2020, when memories of and sentiments about the Black Lives Matter movement were quite strong, and the US Elections were constantly in the news. Combined with the reading, this led many students to look critically at YouTube and empathetically at people who use and are misled by it. For instance, students frequently noted the differences in search outcomes between phrases like "BLM protests" and "BLM riots", and thoughtfully discussed media and public discourse framing of events. Some of these also have national or linguistic characteristics:

> «*After searching for several controversial events and topics in both English and Chinese, I found that results from these queries, which search for the same content but in different languages, reflect a distinct cultural bias. "Black Lives Matter," for one, in English, returns a series of news reports such as "Black Lives Matter protest draws thousands in Brooklyn," but in Chinese, the query results seem to focus more on violent incidents with titles like "* 美国加州游行示威活动演变成打砸抢 *(U.S. California demonstrations evolved into smashing and looting)."*»

Students also found differences that surprised the authors: e.g.,

> «*One query I wanted to check was how Youtube deals with plurals. So the check I gave was the difference between "Mexican" and "Mexicans". My thought was that the latter would produce more objectified or racist content [...].*»

or

> «*[W]hen we type "Donald Trump" into Youtube, we will get videos made by media. However, when we type "Donald J Trump", we will get videos from president Trump's official Youtube channel[...].*»

Overall, students demonstrated remarkable sophistication in their ability to tease out subtle differences in queries that exposed the phenomenon they were reading about (as captured by the reading's title: "the fragmentation of truth"). Many students hit on the concept of confirmation bias. Several students wrote prose about searching "as if they were" some other persona; whether or not they were able to accurately inhabit that persona, we feel that the effort certainly counts, due to its value in AT. However, we also noticed two negatives: (1) Many students changed the *meaning* of their query, got different results, and reported the change, thereby missing the point of the assignment. (2) Few students actually discussed data voids [15], a key concept in the reading.

6.5 Settings

Code	Count (out of 20)
EXP	0
IMP	7
UND	15
REF	2

This was an assignment that lent itself to experimentation: to adopt a persona, use or create an account on a system, and explore it through those eyes. However, the prompt did not call for this. It is telling that none of the sampled students chose to do this in a way that would earn the EXP code.

In this case, the programming component, which explored an alternate data structure representation with no real-world motivation, was perhaps farthest from the C&S component. In keeping with other assignments, we might therefore expect to see students make little connection with implementation techniques. Despite this, a third of sampled student groups chose to comment on implementation (IMP), not tied to the programming component directly but speculating about the behavior of YouTube, Amazon, Reddit, Google Drive, and even data storage in social media companies.

Since the entire point of the assignment was to force students to adopt the perspective of someone else, a high UND count would be unsurprising. What is perhaps surprising is that the count is instead relatively low compared to expectation.

Students commented on algorithmic effects:

> «*Twitter was the source of controversy in late September[.] Normally, the point of the [image-cropping] algorithm is to reduce the size of large images so they don't take up too much space, and so it sought to find the most relevant part of the picture to zoom in on. However, testing showed that when an image showed both a black and white face, the algorithm would default to the white one.*»

and user-interface designs:

> «*Many apps place commonly-used buttons in the lower right-hand corner, where a right-hand-dominant person would be easily able to access it with their thumb. [...] The consequence of this and other righty-oriented tools is that lefties are forced to become more ambidextrous.*»

but also, much more broadly, the ethos of sites:

> «*While LinkedIn promotes the business view as a whole they also create a view of what a normal type of person should be. While society tells people that it is good to stand out, when you look at Linkedin, it's all the same formula, college degree, internship, research, and well known company. [...] For the most part you can not change the order, and they go straight to asking your past and current education, jobs, etc. Linkedin's goal is to help people get jobs and connect with others, but it seems that they do this at the expense of individuality.*»

Multiple students referred back to previous assignment readings (on names, and on the fragmentation of truth). Many students were cautious in their implementation-related statements by guarding them with a conditional: e.g., "assuming that YouTube uses a likes-based algorithm". Much of the thinking was from the perspective of large companies (e.g., Amazon) or countries (e.g., China) to analyze how features would be problematic.

6.6 SafeDesign

Code	Count (out of 24)
EXP	3
IMP	0
UND	23
REF	0

This assignment also featured a split between its programming content (a graph algorithm) and its C&S (understanding abuse vectors). The distance was perhaps a bit less than for Settings because many of these phenomena are tied to social graphs, so they originate from—and perhaps can also be mitigated by—graph algorithms. Furthermore, the presentation of the algorithmic content in class had been from a concrete, socially-driven motivation (rural electrification, which was the reason Borůvka originally studied the problem [4]) rather than as a purely abstract computation. Still, the class presentation and programming task did not have any focus on design for safety.

Interestingly, this slight connection did not have any impact on student responses: no sampled assignment was coded with IMP. (The software itself has relatively few features whose implementation to speculate on.) We also see relatively little EXP, even though the linked video was filled with examples taken from actual systems; presumably the assignment text ("You do not have to test your claims"), the clear demonstration in the video, the exhortation to seek permission, and potential fear of problems if they did experiment on fellow students, precluded any need students felt to experiment on their own.

Unlike in Settings, which expressly asked students to adopt an empathetic position, here students were only asked to identify abuse vectors. Thus, it may appear unusual that so many students achieved an UND score. However, it is worth remembering that the full code for UND includes identifying unintended uses, which was the point of this assignment. Therefore, this high frequency is less surprising (though it is still notable that so many students attained it).

Students observations ranged from the very specific:

> «*The staff have a yellow star next to their names; however, students could simply add a [star emoji] to the end of their last names. It's not exactly the same as the [forum] stars for the staff, but it's close enough to fool most people who aren't looking out for it.*»

to a more careful consideration of roles:

> «*Overall, while [forum] clearly makes an effort to prevent misuse of its platform, it seems to act upon the assumption that abuse comes solely from students. While that's probably true for the most part, instances to the contrary are not unheard of–and are therefore important to consider.*»

to broader statements about platforms and social rules:

> «*[W]ith such a technology that is so geared at designing an ecosystem that best promotes easy communication and effortless lines of contact between people, there are also ramifications of these actions. More importantly, the internet often serves as an amplying vector, providing the space for abusers to have a much larger platform, whereas in normal life, standard social proceedings tend to push such negative people to the fringes. If we want to move forward with the internet in a healthy and positive way, we need to no longer be thinking exclusively about access and free speech, but also consider the ways in which unlimited access to everyone in the world isn't always the top priority.*»

Students also joked (we believe) about being paid for finding security flaws in the discussion forum software.

6.7 Forgetting

Code	Count (out of 20)
EXP	2
IMP	3
UND	15
REF	2

This assignment asked students to speculate about harms, but that did not require having to test any actual systems. Indeed, by now students had played with systems quite a bit (in the course's C&S components itself, even setting aside their other life experiences), so new experimentation was hardly necessary. Without focus on a particular system, there was little cause to speculate on implementation details, either (though in some sense the assignment is deeply about implementations—about how representations inside implementations are reductionist—they were not referring to any *specific* implementation). Unsurprisingly, there is a high incidence of UND, which was mostly fulfilled by students thinking of somebody who doesn't fit into a predefined bin. Finally, we were pleased to see more explicit references to prior assignments.

Interestingly, we see a real drop in response quality on this assignment (independent of the scores, we observed it in the depth and quality of written responses). One plausible cause is simply that students were nearing the end of the semester, when fatigue tends to kick in; this may have been exacerbated by the trying circumstances of taking the course virtually due to COVID-19. We also received (anonymous) comments in an end-of-semester evaluation of course components that explicitly asked about the C&S components. In it, students report that the work near the end of the semester felt repetitive; by this point they had gotten the general messages of thinking about the impact of computing on society, which would include the AT elements this paper cares about. The assignment answers are consistent with ennui and going through the motions.

7 DISCUSSION

RQ 1 asks what kinds of AT can we expect of this student population. The readings cover a broad set of issues, and students respond healthily to all of them. Their responses cover such broad terrain as questioning the meaning of "plagiarism detection", understanding filter bubbles, reflecting on different notions of human identity, observing the pernicious nature of computing tools in contemporary social issues, national differences, and abuse vectors.

However, we also find a *lack* of attention to several issues, which came up rarely if at all. Not all of these are directly related to AT narrowly construed, but many of them have implications for or connections to AT:[6]

- accessibility (other than rare comments such as the one about handedness (section 6.5), or in the context of FORGETTING, which explicitly included the topic in the reading material);
- legal and legislative issues (e.g., handling of data and policies like GDPR; penalizing cybercrimes);
- impacts on children;
- environmental impacts.

On the other hand, we felt an excessive discussion of social media and echo chambers. Teasing out the causes—whether our readings, saliency in popular media, or actual student blind-spots—is an important question for future work.

Furthermore, student were responding to concrete prompts. We provided these prompts expecting that overly open-ended questions would be confusing or get only vague answers that, in turn, we would have difficulty evaluating. However, our questions would naturally have limited what students focused on. It is therefore unclear what the impact of more open-ended questions would be; we believe this would be a useful avenue for further investigation.

RQ 2 asks whether students can meaningfully perform AT before they have technical knowledge to build the systems they analyze. The response to this seems to be a resounding yes. In most cases students were reading about or experimenting with systems vastly more sophisticated than anything they knew how to build at that point (or even that many college graduates may not entirely grasp, if we consider the complexity of a search engine). Yet they were able to find interesting behaviors and suggest failure modes for these systems. This suggests that even with small amounts of computing training, students can help the next generation of systems be less socially harmful, whether as in-house employees or as journalists or as independent analysts. This also relates well to Bruner's notion [5] of "readiness for learning", where he believed materials can be taught in an "intellectually honest form" at many ages.

We believe this outcome is particularly valuable (if it carries over) to courses with students who will not go on to study more computing (e.g., courses for non-majors). The activity has two other potentially beneficial outcomes. First, by providing non-programming work, it changes the balance of activity (and perhaps reduces the frustration) in a typical introductory post-secondary course. Second, students who may (however inaccurately) perceive they are "behind" their peers—especially underrepresented students—can shine in activities like this, drawing on their rich social experiences. This may make them feel more valued and give them confidence. We therefore believe these uses deserve careful analysis.

RQ 3 asks about how students do on social versus technical issues. We find that students predominantly focus on social issues. There are relatively few technical observations or even speculations. Only in very specific situations do students focus on technical aspects (e.g., using different character encodings). Given our student population, it would not have been surprising to see much more technical focus. In a way we find their responses heartening, because speculating technically with only limited knowledge could lead them to arrive at incorrect conclusions, especially ones that might give them a false sense of safety.

[6]These comments are based not only on our sample but also from more broad reading of student responses, while grading.

In several assignments, we see a lack of transfer (as noted by the absence of the REF code). Transfer is, of course, difficult in general. Furthermore, we believe our counts may be lower-bounds, because the earlier work could have directly influenced the latter, and students simply failed to *comment* on it (which is especially understandable given our prompt to write concisely). We therefore think more direct ways of measuring transfer would be valuable: are students seeing clear connections between assignments that we see, or not? If they do not, then the odds of the positive findings above having a broader impact immediately diminish.

Near the end of the sequence of assignments, we sense a degree of exhaustion amongst students. By that point, students have gone over many of the salient social issues. Their exhaustion could be viewed as a kind of evidence for transfer: they spot similarities and don't feel like repeating themselves. At any rate, it seems clear that the students hit a point of diminishing returns.

Overall, students show a broad variety of responses. In particular, they do not seem to suffer excessively from functional fixedness. When asked to formulate other uses, misuses, or abuses of systems, they are able to envision a broad range of alternatives (UND). While these exercises were not designed solely around AT, our numbers indicate that it is meaningful to begin students thinking about AT well before an upper-level security course, and in a much broader setting. Indeed, following Bruner's "readiness for learning", it is interesting to contemplate doing these same activities with even younger students, which could tie into broader education in subjects like civics.

8 THREATS

8.1 Threats to Internal Validity

As noted (section 5), all student work was required to be submitted anonymously, to avoid grader biases from affecting outcomes. While this may be effective for code, it is almost certainly less effective for written prose. Choices of vocabulary, language constructs, grammar, and higher-level features may have led readers to, at least subconsciously, infer (correctly or otherwise) at least a group, if not individual, identity of the writer. This in turn could affect how their work was assessed, thereby leading to mis-coding in either direction (giving too little or too much credit).

As section 5 indicates, the coding was done by staff associated with the course. This may naturally lead us to view the course more positively, leading to ascribing more positive codes than might be assessed by an external observer. Unfortunately, inter-coder reliability is not sufficient to overcome this bias, since both coders could suffer from it. Naturally, the authors are conscious of this and have tried to maintain their skepticism about student performance while performing this coding.

Our conclusions in section 7 are based on our analysis of student performance on certain homeworks. This is a subset of all the homeworks in the course; these were chosen because, in the opinion of the authors (indeed, as part of their *design*), these were intended to have AT elements. However, it is conceivable that the other assignments also did, and the authors suffer from a blind-spot; analyzing those may have produced poorer performance. All materials are available from the course home page: https://cs.brown.edu/courses/csci0190/2020/assignments.html

Students were also told (section 5) to be "brief" and "crisp". There may be several more AT-related issues that they may have written about without these exhortations. The C&S components were also only a small part of demanding projects. As a result, we should treat what they wrote as a a lower bound, rather than as a comprehensive reflection of their understanding. Nevertheless, given that they were not given *specific* guidance on what not to write about, phenomena that they missed as a group (section 7) are still worth noting.

8.2 Threats to Generalizability

Due to our physical setting and student population, our study is inherently US-centric. Naturally, the *kinds* of issues students raise are bound to be influenced by culture. Notions like privacy are known to be very culturally charged (e.g., [32]). Even though our class had several international students, especially from China and India, many non-US perspectives were almost certainly lost. Therefore, we need studies from many more cultural contexts. However, attempting a multi-institutional, multi-national study [13] requires care. Unlike, say, simple programming exercises, even the choice of articles is culturally-loaded, and would not make sense across cultures; thus an interesting research question is what it even means to perform such a study across cultures.

However, it is worth remembering that the critical part of study is not *which* issues students raise, but rather *whether* they are able to raise issues at all. That students in different cultural settings might raise different issues is hardly surprising; what matters is whether they are able to start thinking adversarially from early in their computing career. Our paper provides preliminary evidence that they can; that aspect can be studied in various settings with localization.

The low number of underrepresented minority students in this student population means many issues they may have raised probably went unexplored, especially when we consider that similar lack-of-diversity issues plague developer teams as well. It is conceivable that students from families, races, or other groups that are more threatened by computing (e.g., on the receiving end of surveillance, or denied parole [2]) may be more aware of these issues.

It is possible that a group with less computing preparation than this student population would not have fared as well. However, that is not a given: relatively few comments were deeply technical; most were based on students' interaction with computing devices and systems in general (e.g., Instagram) or their awareness of social issues, both of which can apply broadly. Indeed, it would be very interesting to examine how our findings generalize to less prepared populations.

We should also consider the possibility of a Hawthorne effect-like phenomenon, where student performance on AT and other C&S factors may have been positively impacted by the grading process. In particular, because students received (sometimes detailed) written feedback, they may either have been inspired to work harder or may have been alerted to the attention that was being paid to their responses (or both). Removing some of this (level of) attention could result in poorer attention to AT concerns.

COVID-19 impacted the student population in many ways, and very likely affected their responses. On the one hand, many students were taking only one course (rather than four), and may have had much more time to respond; with a more regular workload, their responses may have been of much poorer quality. At the same time, the impacts of COVID-19 and the Black Lives Matter (and similar) movements may have also created much greater consciousness about disparities and threats.

8.3 Threats to Reproducibility

Strictly from a scientific reproduction perspective, findings based on artifacts like search engine results are problematic because search results are customized both by the searcher and vary across time. The former can be mitigated by redoing the searches from several accounts, in Private Browsing mode, etc. The latter are even harder to control because search engines change. Thus, issues that are salient at one point may fade away later. Indeed, publicizing negative outcomes can itself lead to systems changing their behavior, e.g., as when negative findings about gender bias in translation [7] led to translation systems partially improving their handling [20]. As a result, there is value to revisiting some of these searches later to see whether and how they have changed.

8.4 Ecological Validity

Our ultimate goal is not only to have students be informed about these issues, but to act on them. For instance, suppose they design a new AI system that claims to recognize faces. Compared to students who have not been through this paper's educational process, will these students:

- Recognize up front the harms that such a system can cause?
- If it is implemented, be more likely to test it on diverse populations?
- And ultimately, speak up about the harms they identify?

Similarly, would they produce more secure systems through a combination of better design and superior testing? We cannot easily answer such questions through our current framework.

9 CONCLUSION

In this paper, we have examined introductory post-secondary students' AT ability in an introductory course. Our definition is not narrowly focused on technical cybersecurity but rather includes behaviors where one can adopt the viewpoint of others—which can be put to adversarial use, but can also be used empathetically. We intend to use this as a baseline measure that can be used to evaluate student development longitudinally.

We find that our student population is able to engage in this activity well, adopting a broad range of interesting perspectives. Their responses range beyond cybersecurity to other concerns such as machine learning perils. In the process, they demonstrate that they are able to engage with computing material well beyond their current technical knowledge. At the same time, we find various weaknesses in their responses: some issues (often, ones that receive a fair bit of media exposure) arise repeatedly while others are barely touched upon at all.

Overall, we believe this study's results are encouraging. They suggest that AT topics can be touched on starting from early in a (collegiate) education. This has the potential to better inform future technical study, and also serve as a springboard for deeper understanding. Our paper also presents both weaknesses (such as the coarseness of our rubric) and threats that future work should address and evaluate.

ACKNOWLEDGMENTS

This work is partially supported by the US National Science Foundation. We are grateful to the students of CSCI 0190, and to the creators and maintainers of the ResponsibleCS program at Brown. We thank Malte Schwartzkopf and Kathi Fisler for useful discussions. We appreciate the detailed reviews from the ICER reviewers. The second author thanks Shuchi Grover for a provocative conversation on Twitter. Finally, we thank Rob Goldstone for discussions that are always a joy.

REFERENCES

[1] ABET. 2018. Criteria for Accrediting Computing Programs, 2018–19. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2018-2019/, last accessed 2021-03-25.

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. *ProPublica* (2016). https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[3] Ruha Benjamin. 2020. Race to the Future? Reimagining the Default Settings of Technology and Society. https://www.ncwit.org/video/race-future-reimagining-default-settings-technology-and-society-ruha-benjamin-video-playback, last accessed 2021-03-25.

[4] Otakar Borůvka. 1926. O jistém problému minimálním. *Práce Moravské přírodovědecké společnosti* III, 3 (1926).

[5] Jerome S. Bruner. 1960. *The Process of Education.* Harvard University Press.

[6] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Conference on Fairness, Accountability, and Transparency.*

[7] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186. https://doi.org/10.1126/science.aal4230 arXiv:https://science.sciencemag.org/content/356/6334/183.full.pdf

[8] Lena Cohen, Heila Precel, Harold Triedman, and Kathi Fisler. 2021. A New Model for Weaving Responsible Computing Into Courses Across the CS Curriculum. In *ACM Technical Symposium on Computer Science Education.*

[9] danah boyd. 2019. The Fragmentation of Truth. https://points.datasociety.net/the-fragmentation-of-truth-3c766ebb74cf, last accessed 2021-03-25.

[10] Melissa Dark and Jelena Mirkovic. 2015. Evaluation Theory and Practice Applied to Cybersecurity Education. *IEEE Security & Privacy* 13, 2 (2015), 75–80. https://doi.org/10.1109/MSP.2015.27

[11] Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: Simplified data processing on large clusters. In *Operating System Design and Implementation.*

[12] Karl Duncker. 1945. On problem solving. *Psychological Monographs* 55, 5 (1945).

[13] Sally Fincher, Raymond Lister, Tony Clear, Anthony V. Robins, Josh D. Tenenberg, and Marian Petre. 2005. Multi-institutional, multi-national studies in CSEd Research: some design considerations and trade-offs. In *International Computing Education Research Workshop 2005, ICER '05, Seattle, WA, USA, October 1-2, 2005*, Richard J. Anderson, Sally Fincher, and Mark Guzdial (Eds.). ACM, 111–121. https://doi.org/10.1145/1089786.1089797

[14] Kat Fukui. 2019. Designing for Safety: Principles & Practices. https://www.youtube.com/watch?v=5CSQYMOWOtQ, last accessed 2021-03-25.

[15] Michael Golebiewski and danah boyd. 2018. *Data Voids: Where Missing Data Can Easily be Exploited.* Data&Society. https://datasociety.net/library/data-voids-where-missing-data-can-easily-be-exploited/, last accessed 2021-03-25.

[16] Seth T. Hamman and Kenneth M. Hopkinson. 2016. Teaching Adversarial Thinking for Cybersecurity. *Journal of The Colloquium for Information System Security Education* (Sept. 2016).

[17] Rick Homkes and Robert A. Strikwerda. 2009. Meeting the ABET program outcome for issues and responsibilities: an evaluation of CS, IS, and IT programs. In *Proceedings of the 10th Conference on Information Technology Education, SIGITE 2009, Fairfax, Virginia, USA, October 22-24, 2009*, Don Gantz, Ken A. Baker, and Daniel Garrison (Eds.). ACM, 133–137. https://doi.org/10.1145/1631728.1631764

[18] Joint Task Force on Cybersecurity Education. 2017. Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity. https://cybered.hosting.acm.org/wp-content/uploads/2018/02/newcover_csec2017.pdf, last accessed 2021-03-25.

[19] Frank Katz. 2019. Adversarial Thinking: Teaching Students to Think Like a Hacker. In *KSU Proceedings on Cybersecurity Education.* https://digitalcommons.kennesaw.edu/cgi/viewcontent.cgi?article=1092&context=ccerp, last accessed 2021-03-25.

[20] James Kuczmarski. 2018. Reducing gender bias in Google Translate. https://www.blog.google/products/translate/reducing-gender-bias-google-translate/, last accessed 2021-03-25.

[21] Patrick McKenzie. 2010. Falsehoods Programmers Believe About Names. https://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names/, last accessed 2021-03-25.

[22] Arielle Pardes. 2019. This Dating App Exposes the Monstrous Bias of Algorithms. https://www.wired.com/story/monster-match-dating-app/, last accessed 2021-03-25.

[23] Eli Pariser. 2012. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think.* Penguin.

[24] Allison Parrish. 2016. Programming is Forgetting: Toward a New Hacker Ethic. http://opentranscripts.org/transcript/programming-forgetting-new-hacker-ethic/, last accessed 2021-03-25.

[25] Mark A. Runco. 2014. *Creativity: Theories and Themes: Research, Development, and Practice* (second ed.). Academic Press.

[26] Fred B. Schneider. 2013. Cybersecurity Education in Universities. *IEEE Security & Privacy* 11, 4 (2013), 3–4. https://doi.org/10.1109/MSP.2013.84

[27] Bruce Schneier. 2008. Inside the Twisted Mind of the Security Professional. https://www.wired.com/2008/03/securitymatters-0320/, last accessed 2021-03-25.

[28] Robert Sternberg. 1985. *Beyond IQ: A Triarchic Theory of Intelligence.* Cambridge University Press.

[29] Anselm L. Strauss and Juliet M. Corbin. 1990. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory.* Sage.

[30] Edward L. Thorndike and Robert S. Woolworth. 1901. The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review* 8 (1901).

[31] tony rogers. 2018. Falsehoods Programmers Believe About Names – With Examples. https://shinesolutions.com/2018/01/08/falsehoods-programmers-believe-about-names-with-examples/, last accessed 2021-03-25.

[32] Sabine Trepte and Philipp K. Masur. [n.d.]. *Cultural differences in media use, privacy, and self-disclosure. Research report on a multicultural survey study.* Technical Report 2016. University of Hohenheim.

[33] Debora Weber-Wulff. 2019. Plagiarism detectors are a crutch, and a problem. *Nature* 567 (2019). https://doi.org/10.1038/d41586-019-00893-5

[34] John Wrenn, Tim Nelson, and Shriram Krishnamurthi. 2021. Using Relational Problems to Teach Property-Based Testing. In *The Art, Science, and Engineering of Programming*, Vol. 5. Issue 2.