Deconstructing Conceptions of Rigor in Computer Science Education

JAYNE EVERSON, University of Washington
MEGUMI KIVUVA, University of Washington
EMAN SHERIF, University of Washington
ALANNAH OLESON, University of Denver
AMY J. KO, University of Washington

Objectives In calls for excellent and equitable Computer Science (CS) education, the word *rigor* often appears, but it often goes undefined. The goal of this work is to understand how CS teachers, instructors, and students conceive of rigor.

Research Questions: 1) What do CS instructors think rigor is? and 2) What do students think rigor is?

Methods: Using the principles of phenomenological research, we conducted a semi-structured interview study with 10 post-secondary CS students, 10 secondary CS teachers, and 9 post-secondary CS instructors, to understand their conceptions of rigor.

Results: Analysis showed that no participants had the same understanding of rigor. We found that participants had abstract *Principles of Rigor* which included: Precision, Systematic Thought Process, Depth of Understanding, and Challenge. They also had concrete *Observations of Rigor* that included Time and Effort, Intrinsic Drive, Productive Failure, Struggle, Outcomes, and Gatekeeping. Participants also shared *Conditions for Rigor* which included Expectations, Standards, Community Support, and Resources.

Implications: Our data supports prior work that educators are using different definitions of rigor. This implies that each educator holds different expectations for students, without necessarily communicating these expectations to their students. In the best case, this might confuse students; in the worst case, it reinforces hegemonic norms which can lead to gatekeeping which prevents students from fully participating in the CS field. Based on these insights, we argue that to commit to the idea of quality CS learning, the community must discard the use of this concept of rigor to justify student learning and re-imagine alternate benchmarks.

CCS Concepts: • **Social and professional topics → Computing education**.

Additional Key Words and Phrases: rigor, CS education

1 INTRODUCTION

There are a number of global efforts aimed at ensuring that all students have access to computer science (CS) education. Many of these conversations about improving instruction often use the word *rigor* as a necessity for quality instruction that improves students' experiences. For example, Computer Science Teachers Association (CSTA), the organization whose standards are most widely adopted as state standards in the United States, works to "increase the availability of *rigorous* computer science for all students, especially those who are members of underrepresented groups" [2]. The National Science Foundation, which made possible much of the computing education research in the United States, "funds research and development that is building the

Authors' addresses: Jayne Everson, everjay@uw.edu, University of Washington; Megumi Kivuva, megumik@uw.edu, University of Washington; Eman Sherif, emans@uw.edu, University of Washington; Alannah Oleson, alannah.oleson@du.edu, University of Denver; Amy J. Ko, ajko@uw.edu, University of Washington.

ACM 1946-6226/2025/11-ART
https://doi.org/10.1145/3776542

necessary foundations for implementing *rigorous* and engaging computer science in schools across the U.S." (emphasis added) [1]. Additionally, the Kapor Center Culturally Responsive-Sustaining Framework, which seeks to create equitable CS for all students, calls for *rigorous* pedagogy and curriculum [3].

All of these calls mention rigor, yet none of them define what rigor is or what it should look like in CS education. There is no mission statement, no definition, no field-wide resource that defines the term "rigor." Yet, teachers are encouraged to teach rigorously and researchers are seeking to understand rigorous education.

However, if the CS education community does not understand what rigor is in instruction, or if definitions of rigor vary, at best, courses run the risk of communicating confusing practices and expectations to students. And in the worst case, the community risks using rigor as a tool for gatekeeping and pushes students out of the field. It is unclear what rigor looks like in effective CS education, or how concepts of rigor play into equitable instruction. It is also unclear what students must do to achieve a rigorous understanding of CS and demonstrate they have achieved mastery of the subject. It is also unclear if what students think they must do matches what instructors think students need to do.

With such importance placed on "rigor" in standards, funding, and research, with no clear agreement on what the word rigor means in CS education, and so many varied definitions of rigor in different fields, it is important to understand:

- What do CS instructors think rigor is?
- What do CS students think rigor is?

To answer these questions, we conducted 29 semi-structured interviews with 10 secondary teachers, 9 post-secondary instructors, and 10 post-secondary CS students.

2 BACKGROUND

The following section provides background on how the idea of rigor has been used and understood within various disciplines, as well as how it impacts students' experiences. This section starts with the liberal arts and move to the STEM fields and ends with definitions and uses of rigor in CS.

2.1 Conceptions of Rigor in Liberal Arts

To inform how CS education understands rigor, we can investigate how other fields have conceptualized rigor, starting with the liberal arts.

Researchers have found no consensus on how faculty define rigor. One faculty task force identified the "nebulous concept of rigor" as critical thinking, challenge, mastering complex material, time and labor, and credible work in university courses [31]. A follow up study found that faculty thought that upper-level courses had more rigor than lower-level courses and that students gave faculty higher rankings if their courses were perceived to be rigorous [30]. Additionally, when Draeger et al. conducted surveys and focus groups with liberal arts professors to determine what professors thought was academically rigorous, they found four dimensions: active learning, meaningful content, higher-order thinking, and appropriate expectations [14]. Whereas one study of elite post-secondary institutions found instructional practices like active learning, high expectations, and tying learning to prior knowledge played a role in supporting rigor [8]. In the liberal arts, professors held several definitions of rigor.

Students' understandings of rigor are also unclear. When examining student conceptions, Draeger et al. found that they differed from those of the faculty. Students perceived tough grading, workload, difficulty, interest, and interactions with instructors and peers as rigor [15]. A longitudinal study of college students operationalized two concepts of rigor—workload and instructor expectations. Workload improved student attitudes about learning, and instructor expectations improved students' self-motivated learning. [10]. Whereas, another study found that students have figured out how to play the game of 'college management' not through hard work, but through choosing schedules, limiting workload, and interacting with professors: students were able to appear rigorous [5]. Students also held many conceptions of rigor.

Furthermore, some prior work argues that the concept of rigor harms students because the concept is used as a mechanism for gatekeeping. Nelson argues that traditional methods of rigor, like firm deadlines, content, and fairness, were "dysfunctional illusions of rigor" that prevented him from teaching well [34]. Jack and Sathy argued in *The Chronicle of Higher Education* that instructors use rigor as a way to "weed out" students rather than engage in inclusive teaching practices [28], and Gannon notes in another *Chronicle* article that instructors use rigor as "cloaks to hide practices that actually erect barriers to student success" [19]. Cate Denial employs kindness as a pedagogical approach, believing the best in students, and seeking to listen to and meet student needs. Denial argues that education often sets students up as antagonists, seeking to avoid rigorous work, rather than making work relevant and interesting [42].

In summary, prior work in the liberal arts conceptualizes rigor as: critical thinking, challenge, mastering complex material, time and labor, credible work, active learning, meaningful content, higher-order thinking, appropriate expectations, tough grading, workload, difficulty, interest, interactions with instructors and peers, cognitive challenge, workload, firm deadlines, content, fairness, weeding-out, and gatekeeping. There are many competing conceptions of rigor, and in the liberal arts, it is unclear what rigor is.

2.2 Conceptions of Rigor in Science, Technology, Engineering, and Math

In STEM fields, there are additional definitions of rigor and those vary between disciplines. It is important to look to the domain-specific definitions of rigor by examining its use in education contexts to interrogate whether it reinforces hegemonic norms. In an examination of feminist perspectives in engineering education research, Riley argues that rigor is discipline-specific and that it is often a way to assert white male heterosexual privilege. She states "'Rigor' suggests a sense of being physically and/or mentally taxing, demanding." She then goes on to question who the field is making engineering hard for and points out that rigor often reinforces "gender, race, and class hierarchies in engineering," while also making invisible queer, disabled, low-income students [38]. Therefore, it is important to understand how rigor is used in the domain-specific implementations of CS education, to ensure that CS education is not excluding students by relying on rigor.

In math, there are even more definitions of rigor. One study found that math teachers often perceive rigor as higher grade level content in lower grade level classes and difficulty of problem type [11]. Instead of difficulty, other researchers attempted to define the tension between play and rigor because in the context of math, rigorous content is often characterized as all work, with little room for play [46].

Science education wrestles with the relationship between rigor and teacher expectations. A longitudinal study of the implementation of rigorous standards found that standards alone were insufficient to ensure student learning, and that teacher expectations about what students could achieve were also important [6]. Additionally, a study of university biology students found that as students progressed in their coursework, they found cognitive complexity increasingly important. However, they also found that complexity was enabled by pedagogical practices such as alignment between assessment and instruction, active class practices, and faculty support. [47]

In summary, from STEM we have additional conceptions of rigor: physically or mentally taxing, demanding, higher grade level content at lower grade levels, difficulty of problem, all work and no play, standards paired with teacher expectations, cognitive complexity, active class practices and faculty support. Again no clear definition of rigor arises from STEM.

2.3 Conceptions of Rigor in CS

In post-secondary CS education, rigor was often used as justification for curricular and pedagogical decision-making in the literature but was rarely defined. Sometimes it was mentioned in the context of mathematical correctness. While using mathematical games to teach rigor in CS, Ginat leaned on Webster's definition as "strictness". This study discussed rigor as a necessary problem solving notion, and used it to describe how game players examined the invariance of their algorithms [20]. Other times, rigor had an essential role in making students into better problem solvers and programmers, who consider programs broadly while debugging [21]. One study set mathematical tools as one of three goals in programming, "fun, rigor, and pragmatism" [43]. Course designers used the concept of rigor to legitimize and stratify post-secondary coursework. When working to build introductory CS courses, instructors developed a "rigor-first" approach, arguing that students must build a deep understanding of the "proper fundamentals of the discipline, including concepts, vocabulary, reasoning tools, and laboratory experience" [13]. Others argued that making CS learning both fun and rigorous could help retain CS majors [12]. Similarly, Ramnath and Hoover presented a framework for balancing rigor and relevance in which they define rigor in two ways: first, as "the manner in which the methods of the discipline are adhered to" and second, "to denote the formal structures required by the discipline." They proposed to assess student learning to see if a course was rigorous, and survey students to see if the course was relevant, defining rigor as learning [37].

In K-12 CS education, rigor was also used as an undefined aspirational goal. Sometimes, rigor was briefly mentioned in passing. Rigor was named as justification to update content-based secondary standards that "lack rigor" [17], or in preparing students for "introductory programming courses... that requires rigorous logical and algorithmic skills" [4]. Other times it was an undefined requirement, like in a model proposing rigorous and coherent high school CS curriculum in Israeli high schools [25]. In a collection of case studies in a special issue of *ACM Transactions of Computing Education (TOCE)*, Hubwieser et al. emphasized the move from current practices towards rigorous CS concepts and "rigorous academic computing" [27], implying that there are rigorous and non-rigorous ways to teach CS in K-12 settings. The associated North American case study pointed out that Advanced Placement (AP) courses were considered rigorous and that students with higher scores on the AP exam were ready for college-level work [22]. In a following special issue of ACM TOCE, Hubwieser et al. recommend factors to "implement rigorous computer science education" without defining rigor [26].

In summary, from the CS field, additional conceptions of rigor are: mathematical correctness, strictness, deep understanding of proper fundamentals, adherence to methods, formal structure of the discipline, justification for course or standard improvements, and requirements.

Prior work on rigor, in and out of computing education, demonstrates that the concept of *rigor* is highly contested, discipline-specific, and lacks consensus. Because there are so many definitions of what rigor is across educational fields, so many vague references to rigor, and because rigor is placed so centrally in many of the field's stated goals for learning and education, it is important to understand what it means in CS education, to understand if there is agreement about its meaning.

3 METHODS

Because we wanted to understand the nuances of the conceptions of rigor that instructors and students held, we elected to use phenomenological research methods, which would allow us to build a deep understanding of participant conceptions, descriptions, and lived experiences around a particular topic, in this case, rigor [35]. The focus of phenomenological research is to capture personal descriptions in order to understand the experience as it is lived. We conducted semi-structured interviews to explore conceptions of rigor with instructors and students of CS across North America. We chose interviews over closed-ended surveys or statistical approaches because it allows the researchers to pursue ideas and stories that arose in conversation, and ask additional questions to better understand and to confirm what participants were saying. It also allows researchers to support triangulation of insights and support participant autonomy over how their data was represented, we also employed member checking techniques [7, 41], reaching out to each participant to confirm that the quotes we chose to represent their perspectives aligned with their understandings of rigor.

3.1 Recruiting

We limited participants to students, secondary teachers, and post-secondary instructors learning and teaching in the United States and Canada because this study was language-specific. Rigor as a phenomenon does not exactly translate into other languages, and since the specific word is used so often in CS in the United States, we narrowed our scope to English instruction in North America.

For post-secondary instructors and secondary teachers, we developed an inclusion criteria that sought anyone currently teaching CS courses at the secondary or post-secondary level in North America. We excluded anyone not currently teaching CS at the secondary or post-secondary level in North America in English. To recruit instructors and teachers we used convenience sampling through the CSTA mailing list and the SIGCSE-members mailing list as well as from personal networks. Through the mailing lists, we directed potential participants to a Google form to collect contact information, experience teaching CS, and courses they were currently teaching. We selected a stratified sample of participants to ensure that the population represented a variety of secondary and post-secondary education professionals. For secondary education, we had participants from private, public, and charter schools in both urban and rural areas. For post-secondary education, we selected participants from large public universities, liberal arts colleges, and community colleges. Additionally we used snowball sampling when participants organically recommended people in interviews.

For students, the inclusion criteria were current students who were enrolled in a CS major or a computing adjacent major, and we excluded anyone who did not take their coursework in English. To recruit students, we used personal networks to reach out through undergrad CS mailing lists at a large public university, a small liberal arts college, and a private university in an effort to find a stratified sample. We offered compensation of $25 gift cards to each participant.

3.1.1 Participants. All participants self-reported gender, race/ethnicity, and any other identities they wanted to share in an optional open-field survey. Participant responses are reported in Table 1. If sections are blank it is because participants did not respond to that question.

Pseudonym	Gender	Race/Ethnicity	Other Identities	Major/Level	Courses
Student 1	Female	Eastern European		computer science and math	intro to object oriented programming, data structures, programming nature, scientific computing, algorithms, intro to AI, computational geometry
Student 2	Straight Man	Black and Japanese		Computer Science	Intro Series, Foundations of Computing, Intellectual Property for Engineers, Society and Technology, Systems Engineering, Compiler Construction, Hardware Software interface, Data Structures
Student 3	Male	Asian		Computer Science	Data Structure, Algorithms, Designs of Programming Languages, Discrete Math, Game Systems
Student 4				Computer Science and/or Mathematics	Object Oriented Programming, Programming Nature, Data Structures, Proofs and Fundamentals (instead of Discrete Math)
Student 5	Male	Asian	International Student	Computer Science	Data Structure, Discrete Math, and other lower-level courses...
Student 6	Female	Turkish		Biology and Computer Science	Object Oriented Programming
Student 7	Female	Asian American		COMP SCI	Intro Series, Foundations of Computing, Data Structures, Hardware Software Interface
Student 8	Male	Asian		Computer Science	Intro Series, Web Programming, Foundations of Computing, Software Design, Hardware Software Interface, Linux Tools, Databases, Data Structures
Student 9	Female	Caucasian	Disabled	CS	Intro Series (at Community College), Software Design, Data Structures, Hardware Software Interface, Interaction Programming, Linux Tools, Introduction to AI
Student 10	transgender nonbinary	Chinese American	Neurodiverse (ADHD)	Computer Science and Music Theory	Programming 1, Programming 2, Intermediate Data Programming,Foundations of Computing, Software Design, Data Structures,Systems Engineering, Hardware/Software Interface, Linux Tools
Instructor 1	Male	White		Post-Secondary	
Instructor 2	Male	White (not hispanic)		Post-Secondary	
Instructor 3	Female	caucasian		Post-Secondary	APCSA, APCSP, computer programming
Instructor 4	woman	Asian Indian/other		Post-Secondary	Java, C++, Computers & Society, Intro to Programming (Python), Programming 1, Programming 2
Instructor 5	Female	White		Post-Secondary	
Instructor 6	Male	Half Japanese, half white		Post-Secondary	Programming Languages, Algorithms (and many others less frequently)
Instructor 7	male/man	white		Post-Secondary	Many over the years, but focused on CS1 the last 7.
Instructor 8	Male	White Latino American	I am jewish, an educator, education researcher, a computer scientist, and a software engineer	Post-Secondary	CS1, Discrete math, Databases for Analytics
Instructor 9	Man	South Asian/Indian		Post-Secondary	Web development (front and back-end), mobile application developments, object-oriented programming, projects & outreach, security & privacy
Teacher 1	Female	Black		High School	Adv IT Hon and Coding Fundamentals
Teacher 2	Female	Asian	Able-bodied	High School	AP Computer Science Principles/intro to CS, AP Computer Science A, post-AP seminar
Teacher 3				All	Everything taught at Community College, run a free afterschool program for girls throughout Montana
Teacher 4	I want to identify as non-binary[1]	White	No	High School	Elementary specials for grades 3-5, 3 versions of year-long intro CS, AP Computer Science Principles
Teacher 5	male	Filipino/ German	Engineer/Educator	High School	AP CS-A, AP CSP, ECS
Teacher 6	Mostly Male Identifyied	White / Asian	Queer, Slighty Neuro-Diverse	High School	Ap Computer Science Principal, Ap Computer Science A, Game Design, Game Programming, Cyber Security, Cisco Networking, Web Design, Special Projects, Data Science
Teacher 7				High School	AP Comp Sci, Comp Sci Principles, Intro to CS, Web Design . . . . and so on
Teacher 8	male	white		High School	AP CS A, CS1 Python, CS2 Java, CS3 Java
Teacher 9	Female		Married to a man	High School	Intro, Video Game, AP CS A, AP CSP
Teacher 10	Female	White		High School	Since 1984 .... all grades .... since 2016 ECS, CSP, AP CSP, AP CS A, Advanced Topics in CS

[1] Participant also included "since I believe that gender is a social construct, although it feels disingenuous to identify as anything other than a man, as that is how I've lived and been treated the majority of my life"

Table 1. Participants self-reported gender, race/ethnicty, and other identities with the authors, and shared the major and cs courses they had taken as students or level and courses they were teaching as instructors.

3.2 Positionality and Reflexivity

As authors, we approached this work with experience learning and teaching CS in formal and informal settings. Through this work, we seek to help make CS a more inclusive place with an inclusive culture. We believe that part of this shift must integrate interrogating the terminology used to shape pedagogical decisions within the field. Within our respective subdisciplines, we have each experienced the term "rigor" being used as an excuse to avoid making curricular and pedagogical changes that would center students' needs and experiences. We've also seen it used as justification for why content is taught a certain way in a variety of learning settings. Thus, for this work, we approached the analysis and the word "rigor" with skepticism. When discussing the initial project, none of us had clear definitions of the term, which sparked our interest in better understanding the conceptions held by instructors and students. Broadly, we also believe that learning computing should be a joyful experience.

These views risked influencing how we viewed and judged respondent's responses, and so in considering their ideas, we sought to think about the systems, experiences, and institutions that might have led respondents to those beliefs, rather than examining them judgmentally or as a reflection of individual character.

3.3 Interviews

The first author conducted all the interviews over synchronous video chat (Zoom). The interviews lasted between 30 minutes to an hour and followed a semi-structured format drawing from the following questions.

- What are some things that might be important in (your) students' learning?
- When you think about CS teaching (taking CS classes), how do you think about rigor?
- How do you think your colleagues (instructors) would define rigor?
- Do you think it's possible to teach rigor?
- How do you observe rigor in students? How do you assess if students' learning is rigorous?
- Can you give me an example of rigorous work?
- Can you give me an example of work that does not demonstrate rigor?
- Could you give me an example of rigor in another domain? Would you define rigor differently in another domain?
- What would your definition of rigor be?

The questions were developed through a series of iterative pilot interviews and the first author worked with the research team to refine the questions until it was clear the participants in the pilot interviews were articulating their conceptions of rigor. Transcripts were generated by Zoom and then verified, cleaned, and de-identified by the first, second, and third authors.

3.4 Analysis

We adhere to Hammer and Berland's perspective on qualitative analysis [23], which positions qualitative coding as generating interpretative claims about data for later testing, rather than structured data for quantification. Therefore, rather than reporting inter-rater reliability analyses and quantities, we share here our analysis process and the interpretative disagreements that emerged in building a shared interpretation.

The first, second, and third authors identified significant statements by identifying sentences, thoughts or quotes that provided insight into participants' experiences of rigor following phenomenological research principles [35]. Since phenomenological research seeks to study participants' lived experience of a phenomena, and find commonalities across those experiences, in this work it is important to acknowledge the embedded nature of the researchers [16]. The authors strove to acknowledge and "bracket" our preconceptions of the results, and our lived experiences that informed our understanding of the data as we analyzed it [45].

After we identified the significant statements, we printed them and all authors came together to physically affinity diagram the significant statements. To do this, we grouped the statements and quotes on the walls in a large room. Once we grouped the statements, we selected a title for each grouping and inclusion as well as exclusion criteria for that group to ensure we had team consensus on what each grouping entailed. This allowed for deep discussion on interpretive disagreements.

The analysis team then reorganized and re-coded significant statements that had no group. Finally, to ensure clear groupings, we removed the title from each group of quotes. We then re-read each group of quotes to ensure that they belonged together and re-titled that grouping. In the next round of analysis, we verified the code for each significant statement. We then identified additional codes for the statements. At every step of the way, we collapsed codes until it was clear that if we collapsed these groupings further, we would lose the nuance of the conversations we had with participants.

There was significant agreement with the authors' understanding of the data. Some disagreement did arise during the initial affinity diagramming as authors debated whether participants' thoughts about effort and time should be independent categories and ultimately, collapsed them into a single effort and time grouping. Authors also debated the difference between post-secondary instructor expectations and those of state or school standards, ultimately deciding that those two entities were separate, and they emerged as independent concepts.

The final step of our analysis process was to member-check results with participants. We emailed each participant with any quotes we were attributing to them and the context in which we were sharing their quote to ensure that our analysis represented their intent. At the time of submission, the majority of participants responded (21/29), sharing that the quotes accurately represented their sentiments. Of those participants who responded, a few (4/29) requested slight grammar or punctuation clarifications.

4 RESULTS

We observed fourteen distinct conceptions of rigor, which were organized into three larger categories as seen in Table 2. The first category was *Principles of Rigor*, the abstract properties of rigor that participants shared. These conceptions and ideas were fairly abstract and hard for participants to concretely articulate. The second category was *Observations of Rigor*, which are empirical observations that instructors, teachers, and students used to describe rigor, including how it manifested concretely in work, students, and other areas. The third category was *Conditions for Rigor*, which were learning conditions or environments that were necessary for rigor to exist. When we compared these conceptions across participant groups, we noticed larger trends as seen in Figure 1.

		Student 1	Student 2	Student 3	Student 4	Student 5	Student 6	Student 7	Student 8	Student 9	Student 10	Total Student
Principles	Precision	☑	☑	☐	☑	☑	☐	☑	☐	☐	☑	6
	Thorough & Systematic Thought	☐	☐	☐	☑	☐	☐	☐	☐	☐	☑	2
	Depth of Understanding	☐	☑	☐	☑	☐	☐	☑	☐	☑	☑	5
	Challenge	☑	☐	☐	☐	☐	☐	☐	☐	☑	☑	3
Observations	Time & Effort	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	9
	Intrinsic Drive	☑	☑	☑	☑	☐	☑	☑	☐	☐	☐	6
	Productive Failure	☐	☐	☐	☐	☐	☐	☑	☐	☐	☑	2
	Struggle	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	10
	Outcomes	☑	☐	☑	☑	☐	☐	☑	☑	☑	☑	7
	Gate Keeping	☑	☑	☐	☐	☑	☐	☑	☐	☑	☑	6
Conditions	Expectations	☐	☑	☑	☐	☑	☐	☑	☐	☑	☑	6
	Standards	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	0
	Community Support	☑	☑	☐	☐	☑	☑	☑	☐	☐	☐	5
	Resources	☐	☑	☐	☑	☐	☐	☐	☐	☐	☐	2

		Instructor 1	Instructor 2	Instructor 3	Instructor 4	Instructor 5	Instructor 6	Instructor 7	Instructor 8	Instructor 9	Total Instructor
Principles	Precision	☐	☐	☐	☐	☐	☑	☑	☑	☐	3
	Thorough & Systematic Thought	☑	☑	☑	☐	☐	☐	☑	☑	☑	6
	Depth of Understanding	☑	☑	☑	☑	☐	☑	☐	☑	☐	6
	Challenge	☐	☐	☑	☑	☑	☑	☐	☐	☑	5
Observations	Time & Effort	☐	☐	☐	☑	☐	☐	☐	☑	☑	3
	Intrinsic Drive	☑	☐	☐	☐	☐	☑	☐	☑	☑	4
	Productive Failure	☐	☐	☐	☐	☐	☐	☐	☑	☐	1
	Struggle	☑	☐	☑	☐	☑	☐	☐	☑	☐	4
	Outcomes	☑	☐	☑	☑	☑	☑	☑	☑	☐	7
	Gate Keeping	☑	☑	☑	☐	☐	☐	☐	☑	☐	4
Conditions	Expectations	☑	☑	☑	☑	☑	☐	☐	☑	☑	7
	Standards	☐	☐	☐	☐	☐	☐	☐	☐	☐	0
	Community Support	☐	☐	☑	☐	☑	☐	☐	☐	☐	2
	Resources	☑	☐	☐	☐	☐	☐	☑	☐	☐	2

		Teacher 1	Teacher 2	Teacher 3	Teacher 4	Teacher 5	Teacher 6	Teacher 7	Teacher 8	Teacher 9	Teacher 10	Total Teacher	Overall Total
Principles	Precision	☐	☑	☐	☐	☑	☐	☐	☑	☐	☐	3	12
	Thorough & Systematic Thought	☑	☐	☑	☐	☐	☑	☑	☑	☑	☑	7	15
	Depth of Understanding	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	9	20
	Challenge	☑	☐	☐	☑	☑	☑	☑	☑	☑	☑	8	16
Observations	Time & Effort	☐	☑	☐	☑	☑	☑	☑	☐	☐	☐	5	17
	Intrinsic Drive	☑	☐	☑	☑	☑	☑	☐	☑	☐	☑	7	17
	Productive Failure	☑	☐	☑	☐	☑	☑	☐	☐	☐	☑	5	8
	Struggle	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑	8	22
	Outcomes	☑	☐	☑	☑	☑	☑	☐	☑	☑	☐	7	21
	Gate Keeping	☐	☑	☐	☑	☐	☐	☐	☐	☐	☐	2	12
Conditions	Expectations	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	2	15
	Standards	☑	☐	☑	☑	☐	☑	☑	☑	☑	☐	7	7
	Community Support	☑	☐	☐	☐	☑	☑	☑	☑	☐	☐	5	12
	Resources	☐	☐	☑	☐	☑	☑	☑	☑	☐	☐	5	9

Fig. 1. Each participant's conceptions of rigor sorted by type of participant. There is a heat map of popularity of conceptions at the end of each participant group and overall. Note that no two participants definitions match.

Principles of Rigor: abstract conceptions or properties of rigor	Precision	Algorithmic proof precision, clean code, mathematical exactness
	Thorough and Systemic Process	Systemic structural approach to problem solving that covered all possibilities
	Depth of Understanding	Depth of knowledge to be able to apply knowledge to novel circumstances or real life
	Challenge	Each student challenged at an appropriate level
Observations of Rigor: empirical or concrete observations	Time and Effort	Investing a lot of time or effort on a thing
	Intrinsic Drive	Self-discipline, behaviors of an individual student
	Productive Failure	Willing to fail productively and learn from it, productive struggle
	Struggle	Difficulty of material, often referred to as grind, stress, pain, or misery
	Outcomes	Grades, getting a job, knowing content
	Gatekeeping	People preventing people from learning or continuing in the discipline
Conditions for Rigor: elements necessary for rigorous learning	Expectations	Instructors expectations of student work
	Meeting Standards	State standards, curriculum standards
	Community Support	Scaffolding, online community, and friends you study with
	Resources	Money, resources, or machines

Table 2. Conceptions of rigor participants described, organized by broader groupings, then conceptions and a description of each conception.

4.1 Principles of Rigor

When examining participant responses we observed four abstract *Principles of Rigor.*

4.1.1 Precision. Students referred to proofs of algorithms, mathematical CS concepts, and mathematical CS courses when they spoke about precision. In attempting to describe the abstract precision required for rigor, students mentioned examples of high-stakes algorithms and engineering processes such as bridges and planes because if the work was not rigorous, the consequences would be catastrophic. Student 2 who shared:

> *Rigor is obviously something that is present in math, and is also present in programming, but it's also present in other engineering disciplines, like mechanical engineering, or even especially things like fluid dynamics. These are super complicated things that you really don't want to get wrong. You want to be able to measure even the error on these things. You know they're very rigorous. They're super thorough. You don't want a single part that is ambiguous.*

Mathematical precision and exactness came up often. Both students and professors pointed to mathematical proofs and proofs of a program as a means to explain the abstract concept of rigor.

When instructors and teachers shared their perceptions of precision and thoroughness, they gave examples of code that passed tests and considered all exceptions, algorithms that were mathematically provable, and tracing program execution; where results of the precision gave exactly what was required. Instructor 6 called it the *"opposite of sloppiness,"* and Instructor 8 tied mathematical precision to the systematic thinking required to write code in CS and justified the use of autograders as a way to check for precision:

> *You absolutely can teach students to do rigorous work and to value rigorous work... your code should be provably correct. You should be able to show me how it's provably correct. You should write a proof of your code. There are so many contexts where we do want that to happen... When I think about a student behaving rigorously, or writing rigorous code, I think about an engineering process that is less likely to contain mistakes about code that has been checked and rechecked, that has been tested or proven.*

Completeness and thoroughness were part of *Precision* for Instructor 8. Rigor meant students were to know how to write the right code, prove it, and plan for every possible exception.

4.1.2 Thorough and Systematic Process. Most teachers and instructors expressed the importance of critical and systematic thought required to approach, break down, and solve problems. When instructors talked about rigor, they shared how important it was to use systematic processes to approach a problem. Some instructors used systematic repeatedly. Instructor 9 shared:

> *Rigor: meaning can they learn to systematically think through and and solve problems, so can they systematically learn to read and write code and systematically solve a problem.*

Instructor 1 expounded on that need for thoroughness and thoughtfulness when approaching problems, specifically proofs, and that one approached a proof with systematic thought, that considered all possible scenarios and did not skip possibilities or steps:

> *My definition of rigor is 'thorough' and 'thoughtful' and such like that. And rigorous coursework is coursework that hits all the things it claims to hit or assesses all the things that claims to assess, or exercises in the students all the skills, that it claims to exercise.*

If students did not learn the systematic thinking to thoroughly approach a problem, they had not learned.

Teachers' conceptions tended to focus on the steps of teaching students to think critically and independently. When teachers talked about this conception of rigor, they shared how they taught problem-solving strategies. They did this to help students learn to work independently and solve problems on their own. Teacher 6 shared the importance of teaching students to recognize the smaller components in order to solve bigger problems:

> *Here's a problem in front of me. How do I break it down into bite-sized chunks and then try to use the computer to help solve that problem?*

Only one student mentioned the process, even then as an analogy. Student 3 compared the need to shift the way they thought about approaching problems to therapy, sharing:

> *I think it makes you think of therapy. It's like reshaping the way that you think about a situation or a problem, so that you can handle it better or handle it in a different way than you might initially have, that would have been less successful. So in that sense, I think you can also be taught to deal with rigor.*

When participants described *thorough and systematic processes*, they described processes that lead to thoroughness that produced precise results; this conception seemed closely related to the previous conception of *Precision*.

4.1.3 Depth of Understanding. Many participants shared that rigor meant a depth of understanding in order to use that knowledge in new circumstances, or to apply it in novel circumstances. Students needed to have deep enough understanding of principles so that they could solve a problem they had never seen before. This seemed to participants to be a property inherent in some students as opposed to processes or behavior.

When students shared about rigor in CS, they did not describe the act of programming as the end objective, rather they viewed programming as a method to learn. Student 4 put this sentiment succinctly when they said, *"You learn to program, and then you program to learn."* Student 4 shared this while describing their favorite CS class. They did not learn any new programming concepts, but they were using the concepts and knowledge they had in novel ways to explore simulations of physics and in machine learning. Students often shared examples of using CS concepts to make things that they cared about *because* they had learned rigorously. They were expressing ownership of the things they were learning, rather than learning concepts because they were required to. Student 10 contrasted their opinion of learning with what they perceived professors thought learning was when they shared:

> *Now that you've taken this exam you got like a 100 or whatever: that's [considered by professors to be] progress. I think it's more the purpose of gaining knowledge is so that you can use it, not to just gain the knowledge with it... If you feel that you're enabled to create something or build something, or use the knowledge that you've learned in some way.*

Students described rigorous programming as a means to an end; using those skills to do something that mattered, not just for the sake of writing excellent code.

However, when instructors shared about this conception of rigor, it differed from these student conceptions. To instructors, rigor meant applying knowledge to questions that students had not yet been asked, often in the context of assessment. They shared that they wanted students to be able to apply these skills to novel questions beyond the scope or context in which they were taught. Instructor 2 shared:

> *Rigor in that context would be, it's really going to be something new ... [the solution]'s gonna have a twist [on the material] that you have maybe not seen before.*

Other instructors shared that they carried a responsibility to prepare students for the next stages of life, whether that was as a software engineer, a researcher, or some entirely different pursuit, and mentioned tests or exams with novel questions as a way to ensure that students had that *Depth of Understanding*. This *Depth of Understanding* would be required for transfer, or moving knowledge to future contexts.

Teachers conceptions seemed to fall in between instructor and student conceptions. When talking about rigor, teachers mentioned making sure students were not just following step-by-step instructions or memorizing content, but independently applying skills in new situations. Teachers shared that they found that projects were ways students could demonstrate rigor. Teacher 9 shared: *"That would be my favorite kind of rigor– it's where you're really putting it all together for yourself in a project."*

Even though the method for applying knowledge varied, students, instructors, and teachers all shared that it was important to have a *Depth of Understanding* in order to be able to use these CS skills they learned in novel settings: for instructors it was in assessments, for students it was through something they cared about, for teachers projects seemed to be the best option. *Depth of Understanding* also meant students were empowered to use these skills and take it one step further and create their own things that they cared about. Participants described a level of autonomy that allowed students to create new things. When students shared about this conception, it was deeply tied to self-directed work and joy.

4.1.4 Challenge. When participants discussed *Challenge*, they focused on challenge for an individual student, sharing that what would be rigorous for one individual, might not be rigorous for another individual, but that *Challenge* was required for rigor and that level of challenge varied between students.

Students shared individual perception of levels of rigor in classes and what might be rigorous for one student might not be rigorous for another student, and that one class might be rigorous for one student and not for another student. Student 10 explained:

> *I feel like rigor does vary, but I think it's very personal in how it varies. There's a lot of different factors, but I think generally it depends on the person.*

Instructors shared that they needed to provide *Challenge* for their students, but that they needed to provide different levels of challenge for different students, and that rigor was not a fixed point, but was rather something that was different for each student. Reflection, and students' self-awareness were often part of the discussion of *Challenge* Instructor 4 saw rigor as *"trying, trying trying, and pushing your limits"* and that some students would turn in completed assignments early, while other students who might have less experience, would be working just as rigorously, and would need more time and support in office hours to grasp a new concept.

Instructor 5, who had both secondary and post-secondary experience teaching CS, shared that they worked to make sure each student was able to reflect and select the level of challenge they needed:

> *If I'm doing an activity in class... I am giving somewhere between two and four options of varying difficulty, so that students can practice at a level that feels comfortable, but challenging to them ... I just don't think that giving every student the same assignment could ever be rigorous for everybody.*

They went on to discuss what challenging students might look like in a K-12 AP CS class, where they had witnessed teachers and instructors who would over emphasize the need for everyone to get the highest score on the end of year national AP exam. Instructor 5 then pointed out that all students would benefit from taking CS whether or not they even took the final exam.

When talking about *Challenge* and rigor, teachers focused on centering student needs in this perceptions of rigor. Teacher 6 stated:

> *I also think of rigor almost also on an individual level. Where are you at as a learner? And what is rigorous for You may not be exactly rigorous for somebody else.*

Additionally, teachers mentioned the importance of providing multiple ways for students to engage with and represent their learning.

4.2 Observations of Rigor

While the conceptions of rigor in the last section appealed to abstract principles, some of the conceptions of rigor participants observed were more empirical in nature. Participants shared six such distinct conceptions. These observations came from both teachers and instructors of their students and from students of their peers. Some of these insights overlapped, but had they been collapsed, they would have lost important nuance and salience.

4.2.1 Time and Effort. Often participants observed rigor as the amount of *Time and Effort* an individual would invest into learning a subject or a concept. Students', instructors', and teachers' observations of this conception varied.

Students would often discuss devotion to CS and the willingness to make sacrifices of time in order to succeed. Student 10 shared the experience of deciding when to get their work done on a weekend, and having to choose between work and family obligations when they said:

> *How long do you spend thinking about the homework before you can actually do the homework? Because there's a lot of like pre-planning. I gauge rigor where it's like like, Can you take a break from what you're doing? Do you need to be thinking about your class every single day?*

Students observed a discrepancy between the *Time and Effort* that professors anticipated would be necessary to complete assignments and their own experiences regarding the actual time and effort required. Student 9 shared some of those expectations mismatches when describing how to learn a new concept, and the professors expectations that they would invest the time to find external information to complete class work:

> *I think there is a disconnect between what a professor attributes as appropriate time to spend on an assignment versus what students are actually spending on that same assignment. I've personally seen it come as a surprise when they find out that a question they thought would take students a half hour is taking students 5 hours. So they probably are thinking. 'Oh, yes, the rigor of this assignment. Yes,*

> *this is rigorous, 30 minutes', and the student is looking at this going, 'This is hour 4. I am like hitting a wall. I don't know what to do.'*

When instructors shared these observations, they focused more on the effort required to learn a new thing. Some instructors shared that they expected students to take initiative to put in effort to learning a thing. Instructor 9 shared, *"The word rigor makes me want to think about the effort that somebody would have to put in to learning or to develop something."* Other instructor shared stories of their peers who had invested into making the perfect class that would cover all the material, but that there was no way an average student would have the time to actually succeed in such a course without making significant sacrifices.

When teachers spoke about *Time and Effort*, they would refer to "hard problems," and the ways students would work with those problems, in particular the effort required to learn a new concept and engage with it. Teacher 5 compared the time and effort required to learn CS to the physical fatigue of a strenuous workout:

> *That's why everybody takes computer science. They either love it because they're up for the challenge, or they hate it...You're learning like how, when you go to the gym. You know you had a good workout... when you can't pick up the bar anymore. Picking up your gym bag to leave is like you have to drag it behind you. You know you've had a good workout.*

Students, instructors, and teachers all shared the observation that time and effort were required to learn CS, but their was a nuanced difference in their observations. Students saw the time and sacrifice required to do work well and instructors and teachers focused on the effort required to grapple with a new concept.

4.2.2 Intrinsic Drive. Many participants described that they observed an intrinsic quality, habits, or mindset that students either had or did not have.

Students shared observations of their peers who were always focused and studying, spending all their time reviewing course material, or on researching additional content. Student 7 laid out two ways they observed rigorous students:

> *There's two ways to get good... One is, you're really good at it. Like you're just naturally good. Two is, you work really hard. Kind of try to apply those to computer science or kind of in school in general... I really rely on the second one. I think I'm afraid I'm not that talented at programming. I've relied a lot of the kind of like just grinding.*

Student 7 put themselves in that second category, but perceived that what required much work from them came naturally to a lot of their peers. Other students used language like *"strict and exact"* when observing their rigorous peers' behaviors, or *"passionate"* when observing their rigorous peers' mindsets.

Instructors' observations were split. Some instructors said rigor could not be about students or their qualities, but rather about the complexity of the course. Few instructors thought students attitudes could be rigorous.

Many teachers expressed that they observed rigor to be the drive students brought to the work they did. Several teachers used expressions like, *"extra mile"*, *"discipline"*, or *"pushing one's self"*. Many teachers perceived some students to be rigorous and some students to not be rigorous. Teacher 4 speculated that they thought rigor and intrinsic motivation were correlated, sharing:

> *Perhaps I associate rigor with intrinsic motivation... I guess maybe if you're intrinsically motivated, you want to do it for the sake of itself. And so you want a challenge, and you want to learn and get better.*

Some teachers perceived some students and their academic behaviors to be rigorous– while other students did not practice the same behaviors.

4.2.3 Productive failure. Participants shared they observed rigor when they watched how students recovered from failure in coursework and in learning.

When students talked about failure, they did not focus on the failure, instead they focused on how good it felt to get it right after the failure. Several students also surfaced the tension they experienced that they may not have the space to fail in class. Grades often played a role in the fear of failure– which made it hard for students to take some of the risks they felt were necessary for learning CS. Student 10 shared the analogy of building a sandcastle:

> *I feel like not being stressed is one of the important things, because I feel like learning should be more of a sandbox environment where you can do what you want... If you're building a sand castle, you wouldn't have someone yell at you because it didn't work the first time that you tried to build the sand castle. And so it's the idea of being able to experiment when you learn and also it's not gonna be like. 'Oh, I'm gonna fail you in the class, because, you messed up your first assignment.'*

Few instructors shared that they thought failure was part of rigor. Instructor 3 shared the importance of the reflective process when failing rigorously:

> *Rigor includes challenging oneself to do something difficult. And we're not always going to succeed. So rigor also includes experiencing failure, reflecting on what can be done differently, and trying again.*

When teachers emphasized the importance of failure in rigorous learning, they often included the importance of community support, like working with peers or with a teacher to find mistakes and bugs, and that the school system was not always designed to support that collaborative nature of failure. Teacher 3 was emphasizing how important it was for everyone to learn, not just elite students who would end up in college or with PhDs, when they said:

> *You have to learn certain things, and you don't have the ability to fail like you do when you're in school, and that's the big... readjust from those mistakes and move forward... especially in computer science, because that's what it's all about. You're failing. You're constantly failing. You constantly have bugs, and you constantly made a mistake somewhere. And you're always trying to fix these mistakes.*

Teachers saw the constant iteration on buggy code as a necessary part of rigorous learning.

4.2.4 Struggle. When talking about this observation, students and instructors shared different understandings. Students perceptions were often negative while instructors shared positive perceptions of the struggle.

Students often used words like *"suffering," "miserable," "grind,"* and *"unhappy"* when describing what was required to be successful in a course. Often CS students would describe long hours and in front of a program or in a library, and the sacrifice of a social life or sleep in order to be fully dedicated to finishing an assignment or being a good student. They equated the state of being stressed out with the level of rigor in a course. When asked how they knew a course was rigorous, Student 10 replied, *Stress. Being stressed out... But I think stress is probably the main way that rigor is present, or that I can feel it at least.* And Student 9 pointed to the difference in perception between students and instructors by sharing an example of the contrasts in how professors perceived rigor and struggle and students did:

> *I think sometimes the rigor can be discouraging or straight up, prohibitively difficult. I think that sometimes some assignments and things like that are just straight up prohibitively difficult in how they're laid out...And you'll end up seeing things like a professor announcing, "Hey, so I saw enough of you cheated on the midterm that I'm making a public post about it," and I think that that's an indication that the rigor may be a little bit too high. If people are feeling like they have to resort to something to be able to even pass.*

They then shared that when they observed their peers cheating, it was as a last resort to not fail a class, not due to laziness. They thought that this was a sign that the course was too hard.

Many instructors had a very different perception of this concept and used words like challenging and struggle with a positive view. Although, some instructors reflected about colleagues who made things hard to make things hard. Instructor 3 who shared the tension of creating courses that were challenging because the covered the necessary content, but not difficult just to be difficult:

> *I think that some would say, 'Rigor is making students do hard things.' And I don't think I would disagree with that. But I also think faculty need to deliberately avoid creating assignments that are difficult for the sake of being difficult. What is the learning objective? How can we remove any cognitive load that isn't related to solving the problem?*

Sharing a similar reflection, Instructor 8 spoke about a colleague who was dedicated to providing thorough course content, and would try to fit as much as possible into a term. As a result, students were overwhelmed.

When teachers talked about rigor as a part of course difficulty, they often started by stating what work they thought was not rigorous; e.g. multiple-choice, reading a book and bubbling in answers. And when teachers described rigorous work they used words like *"hard"* and *"struggle"* often with positive connotations. Many teachers thought it was how hard a course was, like teacher 5 who said said, *"Rigor is the level of difficulty."* Then they went on to share ways they supported students to succeed in spite of the difficulty.

4.2.5 Outcomes. Participants mentioned outcomes as a way they were able to observe rigor in themselves (as students) and their students (as instructors). Surprisingly, participants rarely mentioned grades as a goal of outcomes, but did mention things like *"doing well"*, and *"testing"*. Students saw outcomes like gaining content knowledge, reflecting on the new things they were able to do, and passing tests. Student 1 initially referred to rigorous students as doing well in the class, and later corrected herself, *"Rigor means, practice and practice means result, and then I [thought] 'you know, rigor doesn't always mean good grades."'* Student 4 also shared similar thoughts differentiating between rigor and succeeding in a course:

> *I think the classmates that do well are more rigorous. But this implication doesn't necessarily go both ways. If you do well, you're rigorous. But if you don't do well it doesn't always mean you're not rigorous.*

Many instructors shared that they more concerned with outcomes like mastery or making sure students knew the material well. Instructor 4 said, *It's our job to get [students] to demonstrate that competence and to be able to feel confident in what they're doing.* When instructors mentioned grades, it was in the context of confusion around what they communicated to students and about students. Instructor 5 pointed out how grades and identity were entangled, and mused:

> *When we talk about rigor and teachers who maybe think about rigor in a more traditional sense. 'Everybody has to achieve the same level of difficulty in order to be A level or B level...' I wish that our education system was a little bit more flexible in terms of how we think about students, and not thinking about it as like. This is like A level, and if you're not achieving A level, then you're not an A student.*

In general, instructors shared that they wanted to make sure that they were able to help students master content so that students were prepared for careers in CS.

In addition to observations of outcomes like mastery that instructors held, teachers often tied rigorous outcomes to projects that worked, and that students could clearly explain. Teacher 8 mentioned how they knew students had rigorous outcomes and really understood when they explain why broken code did not work.

4.2.6 Gatekeeping. Often students shared stories of having to fit in, or meeting perceptions of others in order to survive in CS. It seemed that in order to be rigorous, or perceived as rigorous, students shared that they felt like they needed to behave like their peers. Student 5 clearly links fitting in to a performance of rigor when they shared:

> *It can prove to be really difficult to survive in this environment... If you don't fit, well, everyone is all about fitting, and everyone is all about the rigor. So everything in the program is about rigor. So it's really difficult to navigate this environment if you're not what they want or what the standard is.*

Several instructors shared things they had witnessed in staff meetings, conversations they had been in, observations of peers, or personal stories of ways that students were discouraged from pursuing CS because they were not *rigorous* enough. Instructor 3 shared one such anecdote about her son:

> *I have a son who is taking a computer science class at a college right now, his first one, and he says, 'Mom, the teacher makes me feel like an idiot.' And I think that happens a lot in computer science and some people call it rigor. It's not.*

Instructor 1 shared that they wrestled with the word *rigor*– as they felt it had some value or proxy for professional decision making about students. But when reflecting they articulated:

> *Rigor is gatekeeping. It's the word we use when we want to pretend there's some actual measurement behind our gut instinct for whether people are succeeding honestly or not honestly, like cheating. But, honestly, like at a level we consider acceptable. Maybe that's it. I want to use that word but... I wanted to have some actual concrete definition. I just can't find it.*

4.3 Conditions for Rigor

Whereas the first and second sets of conceptions of rigor were about the behavior of teachers and students, a third set of conceptions concerned the properties of systems, such as assessment policies, standards, and community resources. This third set is not a set of definitions, but rather a set of necessary conditions teachers, instructors, and students saw as necessary for rigor to exist.

4.3.1 Expectations. This condition for rigor was about instructor expectations for their students. It primarily came from post-secondary students and instructors. Instructors worked to ensure that students met expectations, while students shared they often struggled to understand what those expectations were.

Many students mentioned the need to meet instructors' expectations in order to succeed in post-secondary CS courses. Students expressed frustration that they did not always know what professors wanted, and that they were doing many things to try and meet those standards. In trying to describe professors' expectations, Student 9 shared an analogy:

> *They show you a picture of a chicken, and then they tell you to draw a duck. So you're never gonna get it right. But you're gonna get some of this stuff right, unless you've seen a duck. But they don't show you a duck, so you'll never know. They'll say it goes quack. You're like cool. Let me guess what that means. But there's not enough information to be able to get it right the first time... The level of rigor of a course is always going to be viewed completely differently from like a perspective of a professor/teaching team and the perspective of students taking the course. and it makes it kind of difficult to understand before going into a class, how difficult it's going to be.*

Other students expressed similar frustrations and experiences of trying to figure out if the professor was withholding concept details because that was the best way to learn, or if a professor was unaware that many students needed more instruction to feel successful.

Instructors saw their own expectations as benchmarks for student success. If students were able to meet their expectations, they were learning and succeeding. Several instructors acknowledged that while clear expectations promote student success, they also feared overly high standards might make classes too difficult and hinder student achievement. Instructor 2 reflected on rigor when designing courses:

> *Students will often go to the expectations we set for them... If we set really high expectations for what we expect them to do, they will a lot of times meet that rigorous demand on them... We don't want to*

> *make classes hard for the sake of them being difficult or for creating a reputation of 'This is the class where you learn rigor.' But there is value to having high expectations... I would hesitate to apply rigor to students rather than to the course.*

Instructor 2 felt it was important to delineate between courses and students. Courses could be rigorous, test students over novel content, and have high expectations. However, they did not want to apply the concept of rigor to students. Expectations served as the marker by which students were assessed in undergraduate and graduate CS programs. In secondary classes, the vocabulary differed slightly: teachers often used the language of *Standards.*

4.3.2 Standards. Middle and high school teachers relied on standards as a way to observe and verify that their students were rigorously learning. In this sections, we refer to standards as school, district, or state standards that teachers were charged to help students meet. Only teachers mentioned this observation of rigor.

Teacher 1 shared the importance of really understanding what the standards were, and how to teach them in order to make classes rigorous. They shared that standards ensured rigor in their classroom:

> *I envisioned rigor to be teaching to the standards and then using cumulative assessment rather than evaluative assessment to address a particular objective or problem within the course of study.*

Teacher 6 shared how difficult it was to teach CS because students come into the class with different levels of experience, and different needs in the classroom, but that all students needed to meet the standards by the end of the year:

> *Students might need different things in order to get to that standard. And the way different students show that standard might not be exactly the same.*

Teachers viewed their ability to help students meet standards as a way to benchmark whether or not they were teaching rigorously.

4.3.3 Community Support. Many participants shared that in order to have rigorous learning, it was important to have peers, mentors, and faculty who supported them. They felt rigor did not happen as well in isolation. Students and teachers mentioned these ideas much more often than college and university instructors did.

Students shared stories of making friends with online moderators who taught them tricks about writing code, going to office hours with professors and TAs, and studying with friends. Student 1 shared the importance of community and peers they learned CS rigorously:

> *There will be people grouped together writing it out on boards. We're doing homework together. Because, like I said before, rigor is hard to learn that on your own, [unless] you are guided through, maybe by someone who's better at it, or someone who's also learning.*

When talking about supporting students, teachers often talked about scaffolding and meeting students at their ability level. They shared stories of individual conversations they had with students to assess those abilities. They talked about the care required to not just answer a question but how to help a student figure it out and develop a deeper understanding of CS. This work went beyond teaching just content, and meant teachers and community members were investing in students and relationships to support students. Teacher 1 shared about the support they needed to provide their students:

> *Rigor has to be relevant. It has to be scaffolded for whatever learner that is present in your class. I can't be just teaching to the top ten or twenty percent, but about the students who fall in the middle, what about other students who, you know, need additional support.*

Similarly, Teacher 8 shared that you could not separate rigor from meeting a student where they were to empower them, which require teachers to know each student well enough to know where each individual student was in their level of understanding as well as their emotional state. They thought that a classroom community was required to do so when they shared:

> *I think sort of rigor in teaching is being systematic about what information students need, at what time, in order to feel empowered and approach new challenges creatively without becoming overwhelmed.*

4.3.4 Resources. Resources like access, time, money, and computers were elements that were part of rigorous learning. Students, instructors, and teachers all shared examples of systemic barriers that were tied to a lack of resources that slowed them down or prevented them from learning and teaching rigorously.

When teachers spoke about *Resources* needed to support rigorous learning, They quickly pointed out lack of resources like time and money. Teacher 6 expanded on how these resources interfered with teachers' ability to teach rigorously, sharing:

> *I think that they think that rigor is like making sure that students are challenged appropriately. I think they may have a harder time of essentially quantifying what that is for making sure that it's being applied to everybody. And I think part of [what makes that hard] is also just systemic barriers or systemic racism and ableism, and all that sort of stuff like we're in a system that kind of automatically. It pushes people out of rigor.*

One instructor mentioned that due to limited public school resources and lack of access some students came in with no exposure to CS, pointing out that it was not quite fair to require the same level of rigor as those who had been programming for years. Reflecting on the societal inequities that slowly built up over the course of his schooling experiences, Student 2 shared that they did not have CS courses in their high school before sharing:

> *There's all sorts of directions or sources of rigor that kind of create this web of... Not only is it academic, but it's also societal. There's like a huge huge web that feeds on itself on just multiple sources of little little sources and big sources of rigor that create this almost perfect storm of being able to have such a rigorous system in place.*

Other students shared how they had the resources that enabled early access to CS through parents, friends, or games that allowed them to pursue CS rigorously from an early age, and recognized that as a privilege.

5 DISCUSSION

Our results show that over the course of twenty-nine interviews, 14 conceptions of rigor and no clear consensus on the definition of rigor in CS education emerged. Participants discussed many different **Principles of Rigor** which included *Precision*, *Systematic Thought Process*, *Depth of Understanding*, and *Challenge*. Additionally, participants shared **Observations of Rigor** which included *Time and Effort*, *Intrinsic Drive*, *Productive Failure*, *Struggle*, *Outcomes*, and *Gatekeeping*. Finally, participants shared **Conditions for Rigor** which included *Expectation*, *Meeting Standards*,*Community Support*, and *Resources*. No two participants' definitions of rigor matched, as seen in Figure 1. Each student, teacher, and instructor we spoke with had a different definition of rigor in the discipline of CS. Instructor 2 pointed out that they observed this confusion in conversations with faculty peers:

> *Rigor is one of these interesting words, right, because people will mean different things when they say it... the word gets mixed up with with a lot of other things.*

Many participants shared that they did not hold a simple definition of rigor, and on instinct, asked to look it up– so as not to give a wrong answer. But, in this study we sought to understand what conceptions of rigor participants held and had observed. Some conceptions of rigor were about teacher behavior, some about student behavior, other conceptions were about student work products, and student knowledge. While it may not have been clear what participants thought rigor was, it is clear that there was not one definition of rigor, and that an individual's role as student, teacher, or instructor impacted their conceptions of rigor.

In previous work examining rigor, we know of disciplinary specific conceptions from the liberal arts and STEM. In this work, all of the conceptions presented were rooted in CS. To our knowledge, this work is the first seeking to understand what conceptions educators and students hold of rigor with respect to CS. As Riley notes, it is important to understand what each discipline assumes rigor to be [38].

A careful examination of Figure 1 illustrates some of the differences in the understandings of rigor between participant groups. Although there are not exact boundaries, there are some observable trends. Instructors and teachers held more conceptions of *Principles* than students did. Students held strong conceptions of the *Observations of Rigor*, particularly *Time and Effort* and *Struggle*. In fact, each student shared *Struggle* as a definition of rigor. This difference in definitions between instructors and students may mean that the field may not be clearly articulating what rigor is to students.

When discussing **Principles of Rigor**, The concept of *Precision* matched with mathematical conceptions of rigor and with conceptions of correctness [20, 46]. Participants often shared high-stakes scenarios where human lives were at stake. When participants spoke about *Systematic Thought Process* and *Depth of Understanding* they mentioned thoroughness as part of those two conceptions. Thinking through things thoroughly was part of a systematic approach. The concept of *Depth of Understanding* supported work by Ramnath et al. that rigor and relevance were important in student work [37]. Participants shared joy when they talked about the ability to apply knowledge to new circumstances, or when they understood how larger parts tied together. There was a particular pride that participants evoked when they shared about using their knowledge to create new programs, games or things that they cared about.

When sharing **Observations of Rigor**, participants' concepts of *Time and Effort* aligned with theoretical frameworks that perceive rigor as hard work or time [8, 10, 14, 15]. Students shared examples of their misery and grind culture when trying to put in the *Time and Effort* they felt necessary to master material. Teachers and instructors seemed less aware of the sacrifices students were required to make outside of class in order to learn rigorously. Additionally, a few teachers and instructors seemed to perceive *Intrinsic Drive* almost as a personality trait that some people were born with. But, it is important to consider that study skills are learned behaviors, and that not all students have the same levels of executive function (the ability to plan, set goals, and focus) and that sometimes policies can conflate behavior and worth [32].

The concept of *Productive Failure* was difficult for some students because they saw the importance of failure, but felt limited by grades because failure meant lower grades. Instructors hold power over the grades, as they assign them, and did not mention anything about the conflict between *Productive Failure* and grades. We have designed education systems for success, not for failure. But, by its very nature, CS learning requires failure. For example, debugging is a process that is often explicitly taught because code does not work and fails and must be debugged. Our educational systems of deadlines and grades are not always designed to support the iterative process of such learning.

With respect to *Struggle*, there is a broader conversation about power, where instructors might be creating assignments that are "hard," where difficulty without appropriate support and acknowledgment of students' personal constraints (e.g. familial responsibility, financial needs) that affect a student's ability to invest enough *Time and Effort* to meet an instructor's goals– where *Struggle* becomes *Gatekeeping*. In classes, instructors hold a tremendous amount of power over students' experience, outcomes, and futures [18].

Computing teachers and instructors also risk entangling student identity with *Outcomes* like grades, or behaviors like *Intrinsic Drive* or *Productive Failure*. If classroom instructors make assumptions about students' ability to learn based on instructors' own perceptions of what rigor is, they risk excluding students who do not have the same lived experiences as instructors. As instructors, we know what it took for us to be successful, and we might recreate these scenarios– but really, what we're recreating is success for people who have the same life circumstances we do. This risks excluding anyone who holds different identities and or backgrounds.

When discussing **Conditions for Rigor**, participants' ideas of *Expectations* align with the theoretical framework of cognitive challenge to students [8, 10]. Instructors perceived their expectations as a way to push students. However, it is important to make *Expectations* explicit– often, students shared confusion and frustration about what professors wanted them to do, and they would waste time and effort in an attempt to meet unstated expectations.

Community Support and *Resources* highlight educational inequities. Students shared examples of times where they had access to CS in elementary school– which ignited their desire to study CS. Other students shared that their high schools did not offer CS, so the first time they were introduced to computational concepts was in a freshman CS course, among peers who had been practicing these concepts their entire lives. Yet, both types of students were expected to succeed in the same *rigorous* course.

This paper found, similar to the work done by Draeger et al., there is a mismatch between faculty perception of rigor and what faculty communicate to students [15]. In CS in particular, professors are hoping to articulate precision, and students internalize rigor as the need to grind and make sacrifices in order to prioritize CS. Our results not only show that there is no agreement in CS, but that this lack of shared agreement has impacts on how students are trying to learn. Throughout all of these conceptions, we heard participants' stories of teachers and instructors taking a particular notion of rigor legitimacy and applying it to the students they are teaching– legitimizing particular conceptions and behaviors.

5.1 Our Understanding of Rigor

Rigor is a combination of discordant conceptions educators use to try to ensure students are learning. Educators may believe that rigor arises when students learn difficult things so they use difficult coursework as an aspirational goal for their materials. When rigor is the justification for learning, it legitimizes behaviors that are traditionally dominant, conflates meritocracy and dominance, and continue to perpetuate the white masculine behaviors that many students do not perform [38]. Slaton and Riley argue that, "fortitude and suffering are required for success" -which contributes to the continuation of stratification and violence within the field [40]. To survive this use of rigor, one must conform and conforming reproduces the initial conditions of CS which excluded so many [29]— which may explain why the field continues to struggle with representation [33].

The use of the concept of rigor perpetuates the current white masculine hegemonic state of CS. Rigor is in part about the adherence to the discipline and precision [38]. Adherence to the traditions of a discipline means adherence to how it is presented, how it is taught, and how it is perceived by students. Students already come with hegemonic perceptions of tools. Introducing new contradictory experiences (e.g. sewn circuits when teaching circuitry) broadens the perspectives that students hold, and students learn the content better [36]. Until instructors present new tools, ways of knowing, new forms of pedagogy, and content in CS, the field perpetuates exclusive traditions and expectations [9].

Epistemologically, if CS education uses the concept of rigor to legitimize the discipline, we are perpetuating the traditions and expectations that already overrun the field. In doing so, researchers and practitioners neglect other ways of knowing that could benefit both students in their learning and the field for growth and exploration [44]. Centering epistemic narrowness and a cost-effective transfer of content becomes the focus of CS education rather than knowledge and learning [39]. When rigor is centered, CS education risks limiting not only who joins the field, but also the capabilities of the field itself to take risks, to try new things, and to construct new knowledge.

Of course many of the concepts like *Precision*, *Systematic Thought*, and *Depth of Understanding* surfaced in this study are essential to CS. We should not abandon such ideas. However, the term *rigor* is not clearly defined, and reinforces homogeneity which in turn harms the field.

5.2 Moving Past Rigor

There are many options to replace rigor— one argument might be to clearly define rigor for CS, so that each instructor and student is clear about what rigor is, and how to be a rigorous student. Alternatively, computing could entirely reject the concept of rigor in CS education. The first step would be to admit that rigor is not as objective as the field perceives it to be. In acknowledging its subjectivity, only then can CS education begin to construct an understanding of what would benefit student learning [24].

What replaces rigor might make learning hard to control, and it might make it harder to measure, stratify, and rank students. It would require CS education to focus on and prioritize student learning over content presentation. It would require instructors to build trust with students and believe that when students are entering the learning environment, students are there to learn and that studetns want to succeed [42]. It would require systems and institutions to better support instructors and teachers– so that teachers and instructors have the support, resources, and time to build that trust and new curricula. This type of learning would be difficult to scale, and CS education would require more labor of instructors than systems currently allow. For example, instead of 200-student lectures, it might be 20-student classrooms, where relationships, humanity, and trust can flourish and where students are able to create and engage in the joy of learning hard things. We see shadows of this joy in conceptions of rigor in *Depth of Understanding* where students know material so well they own their knowledge and use it as a tool to create things they care about. To do this it might require instructors and institutions to incorporate student voice when creating assignments that matter and defining learning outcomes. It would require an examination of power in the classroom, reconsidering how grades are earned and assigned so that courses are not about competition, but about learning. We also see it in *Productive Failure* when students finally get the code to compile, or finally debug that final flaw, when they have space to make mistakes and try again. Instructors would need to consider how to allow for students to fail before they succeed, failure takes time, and often current course timelines only allow for success.

5.3 Limitations

This study has several limitations. This study was done in North America with English speakers, as the authors spoke English, however this study due to its methods, excluded many perceptions of rigor around the world, and limited the conceptions of rigor to ones that were part of North American academic culture.

Our results also have methodological limitations. Participants volunteered for this study. We recruited them through education specific listservs. All of the teachers and instructors interviewed were generous with their time and thoughts. They cared about their students and the courses they taught and they each shared how they were working to be excellent educators. This is a threat to this study's results as being a quality educator was part of these participants' identities. Additionally, students were all enrolled in CS or adjacent majors, meaning the recruiting methods excluded any students who had self-selected out of CS, had failed out, or had left the CS major.

Additionally, results were also limited to the viewpoints and perspectives of these participants, if other participants had been selected, it is possible that additional or alternate conceptions of rigor would have been identified. And although researchers worked hard to bracket their assumptions, interpretations were potentially limited by these researchers' experiences and opinions.

5.4 Implications

This study extends our understanding of rigor particularly as it relates to CS. The participants considered rigor an abstract concept and articulated that it is hard to know whether or not students are rigorously learning, or applying rigor to their work. Because it is abstract, CS education relies on observations of rigor. The need to perform rigor shows up explicitly in the ideas of *Gatekeeping* where students spoke about the need to fit in or behave like other students behaved which reinforces hegemonic norms as Riley argues [38]. CS educators share opinions of performing rigor so that their knowledge, identity, and occupation are legitimized. Perhaps, like Nelson mentions, these legitimized opinions prevent excellent CS teaching [34].

This study finds that, particularly in CS, no one person can know all of CS, and yet often designers of courses and assignments expect that of students. Educators should be clear about what it means to thoroughly know a topic in CS and work to clearly articulate expectations of students, and work even hard-to-help students meet those clearly defined expectations.

High school teachers are putting forward tremendous effort to teach CS rigorously. But it is clear the field has vast and varied views of what rigor is. This work calls into question learning standards, and what notions of rigor are implicit in our assessments, and what those assessments means. Additionally, this points to a major hole in CS teacher education at all levels– as teachers can not articulate what excellent CS instruction looks like. Finally, this work raises significant concerns about how students come to see rigor, through lenses of suffering and sacrifice.

6 CONCLUSION

This work finds that no two people have the same definitions of what rigor is. Often, rigor is used to decide if students are *good enough* to be computer scientists. Instead, we should discard the concept of rigor as it applies to students, and focus on whether the instruction we are providing is *good enough* to support ALL of our students. We need to explicitly communicate our expectations, standards, and what it takes to succeed to students, and we need to work to make sure that each of our students has the resources to meet those standards. The goal of computing education should be to help students learn. Student success should be at the center. Students are capable of doing hard things, and when they do those hard things we ask of them, we should celebrate that success.

7 ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1539179, 1703304, 1836813, 2031265, 2100296, 2122950, 2137834, 2137312, 2318257, 2417014 and unrestricted gifts from Microsoft, Adobe, and Google.

REFERENCES

[1] [n. d.]. Computer Science is for All Students! - Special Report | NSF - National Science Foundation. https://www.nsf.gov/news/special_reports/csed/

[2] [n. d.]. K–12 Standards. https://csteachers.org/k12standards/

[3] 2025. *Culturally Responsive-Sustaining Computer Science Education: A Framework.* Technical Report. Kapor Center. https://www.kaporcenter.org/culturally-responsive-sustaining-computer-science-education-a-framework/

[4] Friday Joseph Agbo, Solomon Sunday Oyelere, Jarkko Suhonen, and Sunday Adewumi. 2019. A Systematic Review of Computational Thinking Approach for Programming Education in Higher Education Institutions. *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (2019), 1–10. https://doi.org/10.1145/3364510.3364521

[5] Richard Arum and Josipa Roksa. 2011. *Academically Adrift: Limited Learning on College Campuses.* University of Chicago Press, Chicago, IL, UNITED STATES. http://ebookcentral.proquest.com/lib/washington/detail.action?docID=648124

[6] Mary Antony Bair and David Edward Bair. 2013. Failure, The Next Generation: Why Rigorous Standards are not Sufficient to Improve Science Learning. *International Journal of Education Policy and Leadership* 9, 5 (2013). https://doi.org/10.22230/ijepl.2014v9n5a562

[7] Linda Birt, Suzanne Scott, Debbie Cavers, Christine Campbell, and Fiona Walter. 2016. Member Checking. *Qualitative Health Research* (June 2016). https://doi.org/10.1177/1049732316654870 Publisher: SAGE PublicationsSage CA: Los Angeles, CA.

[8] Corbin M. Campbell and Deniece Dortch. 2018. Reconsidering Academic Rigor: Posing and Supporting Rigorous Course Practices at Two Research Institutions. *Teachers College Record* 120, 5 (May 2018), 1–42. https://doi.org/10.1177/016146811812000503 Publisher: SAGE Publications.

[9] Sapna Cheryan and Hazel Rose Markus. 2020. Masculine defaults: Identifying and mitigating hidden cultural biases. *Psychological Review* 127, 6 (2020), 1022–1052. https://doi.org/10.1037/rev0000209 Place: US Publisher: American Psychological Association.

[10] K. C. Culver, John M. Braxton, and Ernest T. Pascarella. 2021. What We Talk about When We Talk about Rigor: Examining Conceptions of Academic Rigor. *The Journal of Higher Education* 92, 7 (Nov. 2021), 1140–1163. https://doi.org/10.1080/00221546.2021.1920825 Publisher: Routledge _eprint: https://doi.org/10.1080/00221546.2021.1920825.

[11] Jon D. Davis, Amy Roth McDuffie, Corey Drake, and Amanda L. Seiwell. 2019. Teachers' perceptions of the official curriculum: Problem solving and rigor. *International Journal of Educational Research* 93 (2019), 91–100. https://doi.org/10.1016/j.ijer.2018.10.002

[12] Mohsen Dorodchi and Nasrin Dehbozorgi. 2017. Addressing the Paradox of Fun and Rigor in Learning Programming. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (2017), 370–370. https://doi.org/10.1145/3059009.3073004

[13] John P. Dougherty and David G. Wonnacott. 2005. Use and assessment of a rigorous approach to CS1. *ACM SIGCSE Bulletin* 37, 1 (2005), 251–255. https://doi.org/10.1145/1047344.1047431

[14] John Draeger, Pixita del Prado Hill, Lisa R. Hunter, and Ronnie Mahler. 2013. The Anatomy of Academic Rigor: The Story of One Institutional Journey. *Innovative Higher Education* 38, 4 (Aug. 2013), 267–279. https://doi.org/10.1007/s10755-012-9246-8

[15] John Draeger, Pixita del Prado Hill, and Ronnie Mahler. 2015. Developing a Student Conception of Academic Rigor. *Innovative Higher Education* 40, 3 (June 2015), 215–228. https://doi.org/10.1007/s10755-014-9308-1

[16] Jakob Emiliussen, Søren Engelsen, Regina Christiansen, and Søren Harnow Klausen. 2021. We are all in it!: Phenomenological Qualitative Research and Embeddedness. *International Journal of Qualitative Methods* 20 (Jan. 2021), 1609406921995304. https://doi.org/10.1177/1609406921995304 Publisher: SAGE Publications Inc.

[17] Barbara Ericson, Mark Guzdial, and Maureen Biggers. 2007. Improving secondary CS education. *Proceedinds of the 38th SIGCSE technical symposium on Computer science education - SIGCSE '07* (2007), 298–301. https://doi.org/10.1145/1227310.1227416

[18] Jayne Everson, F. Megumi Kivuva, and Amy J. Ko. 2022. "A Key to Reducing Inequities in Like, AI, is by Reducing Inequities Everywhere First": Emerging Critical Consciousness in a Co-Constructed Secondary CS Classroom. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1 (SIGCSE 2022)*, Vol. 1. Association for Computing Machinery, New York, NY, USA, 209–215. https://doi.org/10.1145/3478431.3499395

[19] Kevin Gannon. 2023. Advice | Why Calls for a 'Return to Rigor' Are Wrong. https://www.chronicle.com/article/why-calls-for-a-return-to-rigor-are-wrong Section: Advice.

[20] David Ginat. 2007. Elaborating heuristic reasoning and rigor with mathematical games. *ACM SIGCSE Bulletin* 39, 4 (2007), 32–36. https://doi.org/10.1145/1345375.1345409

[21] David Ginat. 2007. Hasty design, futile patching and the elaboration of rigor. *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE '07* (2007), 161–165. https://doi.org/10.1145/1268784.1268832

[22] Mark Guzdial, Barbara Ericson, Tom Mcklin, and Shelly Engelman. 2014. Georgia Computes! An Intervention in a US State, with Formal and Informal Education in a Policy Context. *ACM Transactions on Computing Education* 14, 2 (June 2014), 1–29. https://doi.org/10.1145/2602488

[23] David Hammer and Leema K. Berland. 2014. Confusing claims for data: A critique of common practices for presenting qualitative research on learning. *Journal of the Learning Sciences* 23, 1 (2014), 37–46. https://doi.org/10.1080/10508406.2013.802652 Place: United Kingdom Publisher: Taylor & Francis.

[24] Sandra Harding. 1992. Rethinking Standpoint Epistemology: What Is "Strong Objectivity?". *The Centennial Review* 36, 3 (1992), 437–470. https://www.jstor.org/stable/23739232 Publisher: Michigan State University Press.

[25] Orit Hazzan, Judith Gal-Ezer, and Lenore Blum. 2008. A model for high school computer science education: the four key elements that make it! *ACM SIGCSE Bulletin* 40, 1 (2008), 281–285. https://doi.org/10.1145/1352135.1352233

[26] Peter Hubwieser, Michal Armoni, and Michail N. Giannakos. 2015. How to Implement Rigorous Computer Science Education in K-12 Schools? Some Answers and Many Questions. *ACM Transactions on Computing Education (TOCE)* 15, 2 (2015), 1–12. https://doi.org/10.1145/2729983

[27] Peter Hubwieser, Michal Armoni, Michail N. Giannakos, and Roland T. Mittermeir. 2014. Perspectives and Visions of Computer Science Education in Primary and Secondary (K-12) Schools. *ACM Transactions on Computing Education (TOCE)* 14, 2 (2014), 7. https://doi.org/10.1145/2602482

[28] Jordynn Jack and Viji Sathy. 2021. Advice | It's Time to Cancel the Word 'Rigor'. https://www.chronicle.com/article/its-time-to-cancel-the-word-rigor Section: Advice.

[29] Betsy James DiSalvo, Sarita Yardi, Mark Guzdial, Tom McKlin, Charles Meadows, Kenneth Perry, and Amy Bruckman. 2011. African American men constructing computing identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2967–2970. https://doi.org/10.1145/1978942.1979381

[30] James E. Johnson, James A. Jones, Thomas G. Weidner, and Allison K. Manwell. 2019. Evaluating Academic Rigor, Part II: An Investigation of Student Ratings, Course Grades, and Course Level. *Journal of Assessment and Institutional Effectiveness* 9, 1-2 (Jan. 2019), 49–78. https://doi.org/10.5325/jasseinsteffe.9.1-2.0049 Publisher: Duke University Press.

[31] James E. Johnson, Thomas G. Weidner, James A. Jones, and Allison K. Manwell. 2018. Evaluating Academic Course Rigor, Part 1: Defining a Nebulous Construct. *Journal of Assessment and Institutional Effectiveness* 8, 1-2 (2018), 86–121. https://doi.org/10.5325/jasseinsteffe.8.1-

2.0086 Publisher: Penn State University Press.

[32] Mara Kirdani-Ryan and Amy J. Ko. 2024. Neurodivergent Legitimacy in Computing Spaces. *ACM Trans. Comput. Educ.* 24, 4 (Dec. 2024), 49:1–49:28. https://doi.org/10.1145/3690651

[33] Jane Margolis. 2010. *Stuck in the Shallow End: Education, Race, and Computing* (1 ed.). MIT Press, Cambridge. Pages: xii–xii.

[34] Craig E. Nelson. 2010. 10: Dysfunctional Illusions of Rigor. *To Improve the Academy* 28, 1 (2010), 177–192. https://doi.org/10.1002/j.2334-4822.2010.tb00602.x _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2334-4822.2010.tb00602.x.

[35] M Q Patton. 1999. Enhancing the quality and credibility of qualitative analysis. *Health Services Research* 34, 5 Pt 2 (Dec. 1999), 1189–1208. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1089059/

[36] Kylie Peppler and Naomi Thompson. [n. d.]. Tools and materials as non-neutral actors in STEAM education. *Journal of the Learning Sciences* 0, 0 ([n. d.]), 1–38. https://doi.org/10.1080/10508406.2024.2380694 Publisher: Routledge _eprint: https://-doi.org/10.1080/10508406.2024.2380694.

[37] Sarnath Ramnath and John H Hoover. 2016. Enhancing Engagement by Blending Rigor and Relevance. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (2016), 108–113. https://doi.org/10.1145/2839509.2844554

[38] Donna Riley. 2017. Rigor/Us: Building Boundaries and Disciplining Diversity with Standards of Merit. *Engineering Studies* 9, 3 (2017), 1–17. https://doi.org/10.1080/19378629.2017.1408631

[39] Melody J. Shank. 2000. Striving for Educational Rigor: Acceptance of Masculine Privilege. In *Masculinities at School*. SAGE Publications, Inc., 2455 Teller Road, Thousand Oaks California 91320 United States, 213–230. https://doi.org/10.4135/9781452225548.n9

[40] Amy E. Slaton and Donna Riley. 2024. Rage-ography: Rigor, Anti-wokeness, and Technoviolence. *WSQ: Women's Studies Quarterly* 52, 1-2 (March 2024), 269–289. https://doi.org/10.1353/wsq.2024.a924321

[41] Norman A Stahl and James R King. [n. d.]. Understanding and Using Trustworthiness in Qualitative Research. ([n. d.]).

[42] Jesse Stommel, Chris Friend, Sean Michael Morris, Pete Rorabaugh, Howard Rheingold, Audrey Watters, Cathy Davidson, Bonnie Stewart, and Ruha Benjamin. 2020. *Critical Digital Pedagogy: A Collection*. Hybrid Pedagogy Inc.

[43] Angel Sánchez-Calle and J. Angel Velázquez-Iturbide. 1991. Fun, rigour and pragmatism in functional programming. *ACM SIGCSE Bulletin* 23, 3 (1991), 11–16. https://doi.org/10.1145/126459.126464

[44] Sherry Turkle and Seymour Papert. [n. d.]. Epistemological Pluralism and the Revaluation of the Concrete. ([n. d.]).

[45] Britta Wigginton and Michelle N Lafrance. 2019. Learning critical feminist research: A brief introduction to feminist epistemologies and methodologies. *Feminism & Psychology* (Sept. 2019), 0959353519866058. https://doi.org/10.1177/0959353519866058 Publisher: SAGE Publications Ltd.

[46] Karen Wohlwend and Kylie Peppler. 2015. All rigor and no play is no way to improve learning. *Phi Delta Kappan* 96, 8 (2015), 22–26. https://doi.org/10.1177/0031721715583957

[47] Sara A. Wyse and Paula A. G. Soneral. 2018. "Is This Class Hard?" Defining and Analyzing Academic Rigor from a Learner's Perspective. *CBE Life Sciences Education* 17, 4 (2018), ar59. https://doi.org/10.1187/cbe.17-12-0278