2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
ASEE
Paper ID #39225

# Exposing Early CS Majors to Technical Interview Practices in the Form of Group-Based Whiteboard Problem Solving Activities

**Dr. Edward Dillon, Morgan State University**

Dr. Dillon received his B.A. in Computer and Informational Science from the University of Mississippi in 2007. He would go on to obtain his Masters and Ph.D. in Computer Science from the University of Alabama in 2009 and 2012, respectively. Dr. Dillon is a newly tenured Associate Professor in the Department of Computer Science at Morgan State University. Prior to his arrival to Morgan State, Dr. Dillon served as a Computer Science Instructor at Jackson State University (2012-2013), and a Postdoctoral Researcher at Clemson University (2013-2014) and the University of Florida (2014-2016). His research focuses on human-centered computing, computer science education, social computing, and broadening participation in computing. Dr. Dillon has received >$750k in research funding and awards from external agencies and non-profit organizations, including the National Science Foundation (NSF), the Maryland Pre-Service Computer Science Teacher Education Program (MCCE), and the Collaborative Research Experience for Undergraduates (CREU - CRA-WP). Dr. Dillon currently serves as a Co-PI for the STARS Computing Corps, which recently has been renewed for funding by NSF. He has also conducted a Faculty in Residency at Google during the summer of 2018 to learn more about this company's culture, practices, and to understand the expectations for candidates (e.g. aspiring CS majors) who pursue career opportunities at this company and related prominent companies in tech.

**Abigail Dina**
**Mariah McMichael**
**Theodore Wimberly Jr.**
**Lauren Brown**

Lauren Brown is a Morgan State University Research Assistant majoring in Computer Science.

**Krystal L. Williams, University of Georgia**

# Exposing Early CS Majors to Technical Interview Practices in the Form of Group-Based Whiteboard Problem Solving Activities

**Abstract:**

Upon degree acquisition, computer science (CS) majors with aspirations to pursue a career in industry are faced with the challenge of effectively showcasing their skills to prospective employers and hiring managers. Technical interviews are one approach used by tech companies to comprehensively evaluate a candidate's skillset and preparation for potential job placement in their respective corporations. A CS major's inability to showcase appropriate skills and preparation during technical interviews could result in missed opportunities for a lucrative career in tech. This problem potentially contributes to the high job *demand* in tech and the low *supply* of prospective graduates to fill them.

Establishing ways to expose CS majors to technical interview practices, and effectively prepare them for success is a growing topic that needs more attention. In academic settings, approaches for gauging technical interview exposure, preparation, and impact on CS majors during matriculation are expanding. Yet, the existence of systematic studies that yield critical details about the impacts of such exposure and preparation over time are needed. Likewise, there are a lack of studies that emphasize relative efforts at earlier stages of student matriculation.

This article discusses a study that exposed early CS majors to technical interview practices in the form of interactive whiteboard problem solving at a Mid-Atlantic Historically Black University in the United States. The results revealed that majority of these students completed assigned problem sets successfully while expressing positive perceptions, and adequate levels of comfort during these experiences. However, they also showed adequate levels of anxiety.

## 1. Introduction

Effectively preparing computer science (CS) majors to become proficient practitioners in the field is a challenge. CS is considered a field with one of the fastest growing career paths in the world [28]. Yet, the supply of candidates needed to meet the demands of such growth is relatively low.

Research surrounding CS majors and how to appropriately prepare them for success has garnered much attention [2-3, 10, 14 18, 21, 29]. Yet, student success that is primarily contingent upon feeding the CS pipeline with new majors, and encouraging their matriculation through a CS program to acquire their degrees may not be sufficient enough. Upon acquiring their terminal degrees, another challenge that CS majors must master is the ability to effectively showcase their developed skills in practical and professional settings. For instance, *technical interviews* are a common evaluative practice used by tech-based hiring managers to screen the proficiency of a candidate's computational skills as part of the hiring process [6]. During these interviews, candidates are expected to solve technical problems using either a physical whiteboard and/or a virtual/online-based system. Other criteria that are evaluated during this process are the candidates' ability to vocalize their thought process while solving a given problem (or problems), and the level of confidence they exhibit during these interactions. The overall scope of a technical interview is to give hiring managers first-hand information about their candidates to determine whether they possess exemplary *technical*, *verbal*, and *interpersonal* skills needed to be effective for the intended job position of pursuit [20].

For CS majors, encountering technical interviews as a requirement for internships and job opportunities are becoming more prevalent. Likewise, failure to perform well during these interviews could prevent them for securing internships or even an anticipated job/career after graduation. Based on the aforementioned skills and traits that students are expected to showcase during a technical interview, exposing them to such expectations during their matriculation process as CS majors may be beneficial.

Literature emphasizing the prevalence of technical interviews and the need to increase exposure in CS courses and curriculums is expanding [5, 7, 23, 25]. Our work proceeds to expand upon these prior efforts by placing a direct emphasis on CS majors, who are early in the matriculation process, by exposing them to technical interview practices in the form of interactive whiteboard problem solving procedures that are group-based. Through gauging the students' performances on the administered whiteboard problem solving assessments while using a relative construct seen in professional technical interviews, the objective is to examine *critical thinking skills*, *psycho-social attributes*, *mental and cognitive* states, and *developed computational skills* these students tend to exhibit during this exposure [5]. Moreover, this work allows us to examine potential discrepancies that students are exhibiting during this exposure. The intent is to address these discrepancies accordingly in order to help students improve their skillset, competencies, and proficiencies with technical interviews and computational problem-solving in general.

## 2. Literature Review

### 2.1 Student Preparation & Challenges

As students matriculate through a CS curriculum, they are expected to develop and master computational skills to become proficient problem solvers. When emphasizing expectations in early stages of the CS pipeline, majors are expected to grasp necessary programming concepts and paradigms for success, become adept with employing the appropriate syntax and semantics from the programming language being used in intro courses, regardless of language simplicity or complexity, and exhibit some level of proficiency for effectively operating the programming tool/editor assigned for those courses.

However, mastering these processes have been known to discourage CS majors and cause them to leave CS altogether [15, 34]. Retaining CS majors during matriculation has been a known issue for some time, and has garnered much attention [4, 15, 34]. When the issue of retention is addressed, the next challenge is to equip CS majors with the necessary skills needed to acquire their degrees and become practitioners in the field. It is one thing for institutions and CS departments to primarily focus on getting their students to the finish line of a course curriculum to obtain terminal degrees (Figure 1). It is also important that these students can showcase and transfer their learned skills to professional, practical, and/or career-based settings (Figure 2).

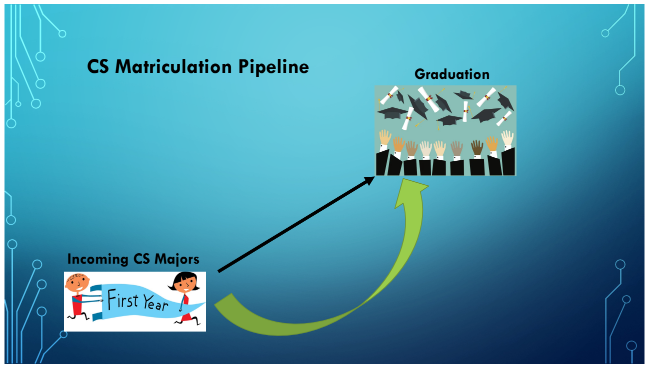

Figure 1: CS Matriculation Pipeline with primary emphasis on graduation/degree production.

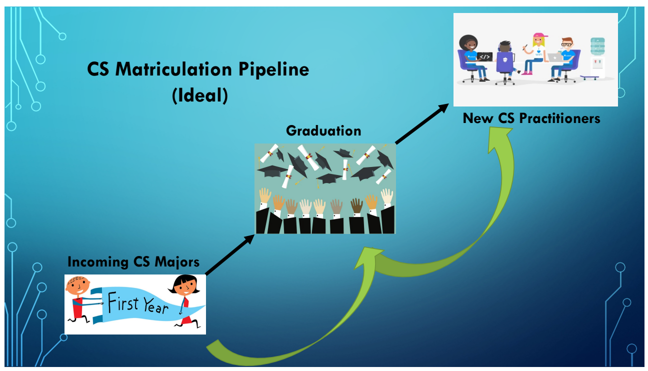

Figure 2: CS Matriculation Pipeline with emphasis on graduation/degree production and career/professional achievement.

## 2.2 Underrepresentation

### 2.2.1 Supply vs Demand

It can be argued that the challenges of retention and job preparation play a role in the low supply of candidates that are available to meet the high demand of jobs in tech. For instance, recent data show that in 2020 there were 400,000 CS graduates looking for jobs in the industry which had nearly 1.4 million computing jobs available. If this candidate to job opportunity ratio were quantified into a monetary value, then there were $500 billion worth of unfilled positions that occurred during that year [31]. This poses the question for how can a sufficient supply of candidates be produced to meet this demand? Such a question places an additional burden on CS and related computer-based departments to increase their productivity of graduates who can fill these positions as practitioners.

### 2.2.2 Minorities in Tech

When it comes to the supply and representation of minorities in tech, this status is much worse. According to the US Equal Employment Opportunity Commission [19], the representation of

African-Americans, Hispanics/Latinx, and Women in the tech industry are disproportionately lower than their Caucasian and Asian counterparts. It has been argued that this current representation is a direct reflection of unfavorable perceptions and stigmas that have plagued the field of Computing for some time as it pertains to race and ethnicity [24]. There have been initiatives by tech companies [27, 30], who are making efforts to address this issue around retention, especially with underrepresented minorities. Likewise, tech companies have begun working closely with minority-serving institutions in efforts to provide insight on the type of computational skills and programming proficiency a student (or prospective employee) must possess for success in these sectors [11, 22, 33]. One anecdotal and common insight from their observations concerns a candidate's ability to exhibit proficient critical thinking skills to solve problems through technical interviews.

#### 2.2.3 Importance of Work

The field of Computing has much potential to provide spaces for everyone to showcase their skills, interests, and contribution to the field [17]. The low supply of candidates along with the underrepresentation of minority groups complicate the supply of practitioners that are needed to support the ever growing and evolving field of Computing. Our work regarding technical interview preparation serves as one effort that is intended to address the low supply of CS candidates as well as minority candidates who are underrepresented in the field. The next subsection provides content on relative initiatives that have been conducted to prepare CS majors for technical interviews.

### 2.3 Technical Interview – Preparation Initiatives

Initiatives to address technical interview preparation for CS majors are expanding. Companies and organizations alike are making resources available for students to prepare for technical interviews [1, 13, 26, 32]. In academic settings, institutions have also begun to expand their resources and/or adjust their CS curriculums in an effort to foster student exposure to the technical interview process [8, 12, 35]. Moreover, academic scholars are now conducting case studies and related interventions to tackle potential challenges that are associated with the technical interview process [7, 20, 23, 25].

#### 2.3.1. Persistent Finding – Performance Anxiety

When observing prior efforts that highlight student performance during mock technical interviews, anxiety has been noted as one persistent attribute that students exhibit during these experiences. For instance, one aspect of Behroozi et al.'s work [7] compared anxiety levels that their participants exhibited while conducting mock technical interviews either in a public setting or in a private setting. It was determined that participants who conducted technical interviews in a public setting exhibited higher levels of anxiety than their counterparts who were in a private setting. Similarly, Hall and Gosha [23] conducted a study that measured the correlation of anxiety and preparation in a technical interview that targeted junior and senior CS majors at a Southeastern Historically Black College/University (HBCU) in the United States. Key information collected during this study were these students' plan of preparation to practice for technical interviews, and whether anxiety played an integral role during their participation for technical interviews. From this work, it was found that anxiety was an underlying factor that could determine a student's overall performance in an interview. It was also concluded that as students become more exposed to technical interview practices their anxiety decreases, while in turn their overall performance increases.

## 3. Method

The objective of the interactive whiteboard problem solving study is to examine the students' ability to conduct critical thinking, verbally communicate their ideas, and create solutions to a given problem. So far, this assessment has been conducted over a span of 3 to 4 semesters ranging from Fall 2020 to Spring 2022. A small portion of this work has already been published by the authors that strictly focuses on its initial impacts during Fall 2020 semester [16]. Newer outcomes regarding this work are discussed in this current article that represent findings that trend over a span of 3 to 4 semesters instead of one.

During each of the targeted semesters, this study included a *PRE* and *POST* assessment to gauge the students' problem-solving abilities at different points throughout a given semester. The targeted participants for this study were students enrolled in either the CS2 or Object-Oriented Programming (OOP) course at a Mid-Atlantic HBCU in the United States. *Just to note: the OOP course was not offered during the Fall 2021 semester, therefore the results and findings for this specific course only reflects three semesters (Fall 2020, Spring 2021, and Spring 2022).*

At this Mid-Atlantic HBCU, CS2 students are traditionally taught intermediate programming concepts and data structures using Python. The OOP students are traditionally taught advanced programming and object-oriented structures using C++. Since these assessments occurred during the height of the COVID-19 pandemic, adjustments had to be made to both courses. Rather than using physical whiteboards as initially planned for this study, the Zoom Video/Web Conferencing system was used as a virtual alternative (Figure 3).

Figure 3: Zoom Video/Web Conferencing System

### 3.1 Whiteboard Assessment Protocol

For each whiteboard problem solving assessments, there were two tasks assigned. During the *PRE* assessment, the students logged onto Zoom and were divided into groups of three and randomly assigned a procedural programming problem to solve from a list of five problems. These groups were then assigned to a Zoom breakout room to solve this problem using a programming tool/editor of their choice. As part of this process, each group collectively had to explain aloud their critical thinking and reasoning for approaching this problem and were encouraged to generate pseudocode that reflected this reasoning. Afterwards, each group implemented their determined solution using a programming tool/editor and given programming language syntax (*Python* or *C++)* to verify that their solution is *correct*.

Table 1 provides a relational table that illustrates how the skills emphasized during these whiteboard problem solving assessments are relative to the ones evaluated during a formal technical interview session. For instance, the students' ability implement code and produce correct solutions are vital components of this whiteboard assessment, which is also employed in technical interview sessions to evaluate a candidate's technical skills and programming proficiency. Moreover, students are required to think aloud during this assessment, which is also used during technical interviewers to gauge the candidate's verbal skills. The group-based aspect of this assessment allows students to develop their soft skills as it pertains to their personal interactions, which is also something that tech interviewers evaluate during their interactions with a candidate. An additional attribute that this group-based whiteboard assessment provides the students is exposure to teamwork/collaborative skills, which is also an integral practice in industry and relative professional settings.

Table 1: Relational Table – Evaluated Skills

| **Evaluated Skills** | ***Technical Interview Protocol*** | ***Whiteboard Problem Solving Assessments*** |
|---|---|---|
| ***Technical Skills*** | Computational/Critical Thinking | Computational/Critical Thinking |
| ***Verbal Skills*** | Thinking aloud | Thinking aloud |
| ***Interpersonal Skills*** | Personal Interaction | Personal Interaction<br>Teamwork/Collaboration |

### 3.2 Setup – For a Given Task

Aforementioned, the students (in their groups) were randomly assigned a procedural programming problem to solve from a list of five problems Each group had 30 minutes to complete the first task (*or Task 1*). After completing *Task 1*, all groups return to the main Zoom room for a brief intermission and to be randomly assigned a different procedural programming problem from the same list of five problems and return to their breakout rooms in Zoom to complete the second task (*or Task 2*). Like *Task* 1, each group had 30 minutes to complete *Task 2*. For both tasks, each group was also given accessibility to record their interaction. Upon completing *Task 2*, the students were then given a survey to complete regarding their experience. To conclude this assessment, each group were required to submit their recordings as a *.mp4* file to Canvas. The protocol for the *POST* assessment was nearly identical to the *PRE* assessment. The only difference was that during *Task 2* the students in both courses, respectively, were randomly assigned an OOP-related problem from a list of four to solve in their assigned groups.

### 3.3 Demographics

The demographic representation for the CS2 course was primarily *freshmen* and occasionally *sophomore* CS majors. There was also a subset of students in this course who are Screen Writing and Animation (SWAN) majors. For the OOP course, the demographic makeup primarily included *advanced sophomores* and *junior* CS majors. For both courses, the ethnic representation of these students was predominantly *Black/African-American*.

**3.4 Data Collection Instruments**

Aforementioned, there were two instruments used to collect the data for these whiteboard problem solving assessments: *surveys* and *.mp4 recordings*. For this article, the results and findings only reflect the students' feedback to these administered surveys. The data collected via the recordings are being cleaned and analyzed. It is the authors' plan to disseminate findings from these recordings in future articles.

**4. Results**

The administered survey for this assessment comprised a series of questions to capture student feedback pertaining to their *prior coding interview experiences, task completion, initial perception, comfort levels, easiest and hardest aspects,* and *anxiety levels.* Each attribute and its finding are discussed in further detail below.

**4.1 Prior Exposure**

During this study, one potential confounding factor that must be accounted for is whether any of these students possessed prior exposure to technical interviews before participating in these two administered Whiteboard Problem Solving assessments. The results revealed that *22%* of the CS2 students, and *30%* of the OOP students, respectively, had prior exposure to technical interviews. Figure 4 provides descriptive details about this finding.

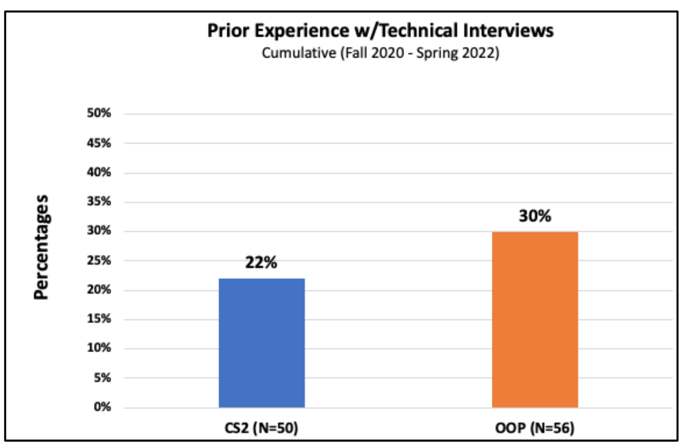

Figure 4: Prior Experience with Technical Interviews

**4.2 Task Completion Rates**

Another aspect of this study was to evaluate whether these students succeeded in completing the given tasks during an administered Whiteboard Problem Solving assessment. With exception to the collective performance of the CS2 students on *Task 2* of the POST assessments, majority of these students (ranging from *60%* to *84%*) successfully completed either Task 1 or Task 2 during a given assessment. It is expected that the CS2 students would struggle on the OOP-based problems during Task 2 of the POST assessment since they lacked adequate exposure to such content. Table 2 provides descriptive detail about this finding (*next page*).

### 4.3 Initial Perception

Whether these students had prior exposure to technical interviews or not, it was also helpful to gauge their initial perception of these Whiteboard Problem Solving assessments. Moreover, it was meaningful to observe whether this perception changes upon further exposure to these assessments. It was found that many of these students' initial perception were favorable: CS2 students (*52%*) and OOP students (*71%*). Likewise, this perception improved favorably over time for majority of them: CS2 students (*77%*) and OOP students (*68%*). Table 3 provides descriptive detail about this finding.

Table 2: Task Completion Rates – Descriptive Data

| Task Completion Rates<br>Cumulative (Fall 2020 – Spring 2022) | | | | |
|---|---|---|---|---|
| | CS2 | | OOP | |
| **Assessment Type** | **N** | **Completion %** | **N** | **Completion %** |
| **PRE Assessment (Task #1)**<br>***Procedural/Functional Programming*** | 50 | 68% | 57 | 84% |
| **PRE Assessment (Task #2)**<br>***Procedural/Functional Programming*** | 50 | 60% | 57 | 75% |
| **POST Assessment (Task #1)**<br>***Procedural/Functional Programming*** | 30 | 80% | 32 | 69% |
| **POST Assessment (Task #2)**<br>***OOP Programming*** | 31 | 42% | 31 | 61% |

Table 3: Initial Perception & Improved Changes

| Initial Perception (PRE) & Improved Changes in Perception (POST) | | | | |
|---|---|---|---|---|
| | CS2 | | OOP | |
| **Assessment Type** | **N** | **%** | **N** | **%** |
| **Initial Perception** | 48 | FR: 52%<br>N: 19%<br>UR: 29% | 56 | FR: 71%<br>N: 13%<br>UR: 16% |
| **Improved Changes in Perception** | 30 | Yes: 77%<br>No: 23% | 31 | Yes: 68%<br>No: 32% |
| *FR = Favorable Response, N = Neutral, UR = Unfavorable Response* | | | | |

### 4.4 Comfort Levels

This attribute examined the students' level of comfort for conducting a given whiteboard problem solving activity and whether this level of comfort changed over time. This was measured using a 10-point Likert scale (1=*not comfortable at all*, 10=*absolutely comfortable*). These scores were then normalized on a scale of 0 to 100, where the *higher* the score the *greater* the comfort level. The results revealed that the students showed an adequate level of comfort upon their initial exposure to a given Whiteboard Problem Solving assessment (ranging from *66* to *77*), which also increased after completing a given assessment (ranging from *73* to *83*). Table 4 provides descriptive detail about this finding. To compare the comfort levels exhibited before and after a given assessment, a series of two-tailed T-Tests were used to determine whether these increases are significantly different. With exception to the OOP students' POST assessment, the T-Tests revealed that the students' comfort level tended to be significantly higher after completing a given assessment ($p \leq 0.01$).

Table 4: Comfort Levels

| **Comfort Levels** | | | | | | |
|---|---|---|---|---|---|---|
| | **CS2** | | | **OOP** | | |
| ***Assessment*** | **N** | **Mean B/A** | ***% Increase*** | **N** | **Mean B/A** | ***% Increase*** |
| **PRE** | 49 | 67/74 | 10.45 | 57 | 73/83 | 13.74 |
| **POST** | 31 | 66/73 | 10.88 | 32* | 77/80 | 3.63 |
| B/A *=Before/After (Assessment), *one student did not provide an answer* | | | | | | |

**4.5 Easiest & Hardest Aspects of Assessment**

To capture detailed experiences from the students during this assessment, a subset of open-ended questions were asked to gather their perception pertaining to the easiest and hardest aspects of this experience. Document analysis [9] was used to categorize these students' responses into five attributes (as seen in Tables 5 & 6, respectively). For both courses, a greater percentage of these students tended to view *Conceptualization/Understanding the Problem* and *Verbal Communication with Assigned Partners* as the easiest aspects of these whiteboard problem-solving assessments. Likewise, students in both courses tended to view *Coding the Problem* as the hardest aspect of these assessments. Tables 5 and 6, respectively provide descriptive detail about the students' view of the easiest and hardest aspects of these whiteboard problem-solving assessments.

**4.6 Performance Anxiety**

Aforementioned, prior work surrounding technical interviews have focused on the dynamics of performance anxiety and how it can play a critical factor in the students' experience in such settings. Our study also examined the potential impact of anxiety and whether it played a role during these students' experiences. This was measured using a 10-point Likert scale (1=*absolutely anxious*, 10=*not anxious at all*). The scores were then normalized on a scale of 0 to 100, where the *lower* the score the *higher* the anxiety. The results revealed that both the CS2 and OOP students, respectively, exhibited adequate levels of anxiety (ranging from *52* to *50*) that remained slightly stable between the PRE and POST assessments. Table 7 provides descriptive detail about this finding (*next page*).

Table 5: Easiest Aspects of Whiteboard Problem Solving Assessment

| Whiteboard Problem Solving – Easiest Aspects | | | |
|---|---|---|---|
| | **Attributes** | **PRE (N=49)** | **POST (N=32)** |
| **CS2** | Conceptualization/Understanding the Problem | 43% | 19% |
| | Coding the Problem | 16% | 19% |
| | Tool Software/Application Usage to Solve Problem | 8% | 0% |
| | Verbal Communication with Assigned Partners | 24% | 44% |
| | Task Itself Was *Not* Easy | 6% | 6% |
| | Other | 2% | 13% |
| | **Attributes** | **PRE (N=57)** | **POST (N=33)** |
| **OOP** | Conceptualization/Understanding the Problem | 32% | 39% |
| | Coding the Problem | 14% | 30% |
| | Tool Software/Application Usage to Solve Problem | 4% | 3% |
| | Verbal Communication with Assigned Partners | 47% | 27% |
| | Task Itself Was *Not* Easy | 0% | 0% |
| | Other | 4% | 0% |

Table 6: Hardest Aspects of Whiteboard Problem Solving Assessment

| Whiteboard Problem Solving – Hardest Aspects | | | |
|---|---|---|---|
| | **Attributes** | **PRE (N=51)** | **POST (N=32)** |
| **CS2** | Conceptualization/Understanding the Problem | 22% | 22% |
| | Coding the Problem | 33% | 31% |
| | Tool Software/Application Usage to Solve Problem | 0% | 3% |
| | Verbal Communication with Assigned Partners | 24% | 0% |
| | Task Itself Was *Not* Hard | 10% | 13% |
| | Other | 12% | 31% |
| | **Attributes** | **PRE (N=55)** | **POST (N=33)** |
| **OOP** | Conceptualization/Understanding the Problem | 15% | 18% |
| | Coding the Problem | 31% | 36% |
| | Tool Software/Application Usage to Solve Problem | 5% | 0% |
| | Verbal Communication with Assigned Partners | 15% | 24% |
| | Task Itself Was *Not* Hard | 9% | 9% |
| | Other | 25% | 12% |

Table 7: Performance Anxiety Levels

| Performance Anxiety Levels | | | | |
|---|---|---|---|---|
| | **CS2** | | **OOP** | |
| ***Assessment*** | **N** | **Mean** | **N** | **Mean** |
| **PRE** | 50 | 52 | 56 | 52 |
| **POST** | 31 | 50 | 32 | 51 |
| **% Increase** | | -3.66% | | -1.73% |

## 5. Discussion

When observing these current results in further detail, it was found that many of these students in both the CS2 and OOP courses exhibited success with completing the assigned tasks during these whiteboard problem solving assessments, even though majority of them lacked prior experience with conducting technical interviews. Furthermore, these students exhibited favorable perceptions towards this style of problem solving, which tended to improve after completing a given assessment. This was also true when observing their comfort levels before and after a given assessment. Based on the students' view of the easiest and hardest aspects of these assessments, being able to conceptualize and understand a given problem along with verbally communicating their thoughts with others were two critical factors that employers and hiring managers evaluate during a technical interview. Aforementioned, majority of the students viewed these two factors as the easiest aspects of these assessments, which only strengthens the validity of this style of problem-solving and initiative. Majority of these students also viewed the actual coding of a problem as the hardest aspect. During technical interviews, a candidate's ability to effectively code an assigned problem is another critical factor that is evaluated. Therefore, this particular aspect of the students' experience may require more attention. When observing performance anxiety, it is found that these students exhibited adequate levels of anxiety which remained stable between both sets of assessments during a given semester. This finding contributes to prior efforts that viewed anxiety as a persistent factor for students during a technical interview-related experience.

## 6. Conclusion

The objective of this article is to disseminate current outcomes pertaining to early CS majors while being exposed to aspects of the technical interview process in the form of group-based whiteboard problem-solving activities. Over a span of multiple semesters, we see collective outcomes that reflect the students' success with completing these whiteboard problem-solving activities, their favorable perceptions and comfort, and notable aspects that they perceive as easy and challenging, whether they have prior experience with technical interviews or otherwise. We are also witnessing some challenges that need to be addressed in order to improve student engagement and confidence while reducing performance anxiety during this style of computational interaction. It will be critical to rectify these issues to help students enhance their competencies, proficiencies, psycho-social skills for conducting computational problem-solving, which could positively impact their successes as they go through official technical interviews.

## Acknowledgements

This current work is funded by the National Science Foundation (award no. *NSF-EDU 2011793*).

## References

[1] AlgoExpert.io: "Ace the Technical Interviews." https://www.algoexpert.io/product. (Retrieved August 18, 2022).

[2] O. Astrachan, K. Bruce, E. Koffman, M. Kölling, and S. Reges. "Resolved: objects early has failed." *ACM SIGCSE Bulletin*., 37, 1, 2005, pp. 451–452.

[3] F. Bailie, M. Courtney, K. Murray, R. Schiaffino, and S. Tuohy. “Objects First-does it work?” *Journal of Computing Sciences in Colleges*, 19, 2, pp. 303–305, Dec 2003.
[4] T. Beaubouef and J.Mason. “Why the high attrition rate for computer science students: some thoughts and observations.” *ACM SIGCSE Bulletin*., 37, 2, 2005, pp. 103–106.
[5] M. Behroozi, A. Lui, I. Moore, D. Ford, and C. Parnin. “Dazed: measuring the cognitive load of solving technical interview problems at the whiteboard.” *In Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results*, 2018, pp. 93–96.
[6] M. Behroozi, C. Parnin, and T. Barik. “Hiring is broken: What do developers say about technical interviews?” *In 2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2019, pp. 1–9.
[7] M. Behroozi, S. Shirolkar, T. Barik, and C. Parnin. “Does stress impact technical interview performance?” *In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 481–492.
[8] A. Benson and I. Tullis. “CS 9: Problem-Solving for the CS Technical Interview.” https://web.stanford.edu/class/cs9/ (Retrieved August 18, 2022).
[9] G. A. Bowen. “Document analysis as a qualitative research method.” *Qualitative research journal.*, 9, 2, pp. 27-40, Aug 2009.
[10] R. Brown, J. Davis, S. A. Rebelsky, and B. Harvey. “Whither scheme? 21st century approaches to scheme in CS1.” *ACM SIGCSE Bulletin.,* 41, 1, 2009, pp. 551–552.
[11] C. Cain, C. Buskey, A. M. Bryant, G. Washington, and L. Burge. “Research Implications of the Tech Exchange: Immersion of Howard University Computer Science and Informatics Students in Silicon Valley.” *Twenty-fifth Americas Conference on Information Systems.,* 2019, pp. 1-5.
[12] Caltech.edu. “CS 11: Interview Preparation Track.” http://courses.cms.caltech.edu/cs11/material/interviews/. (Retrieved August 18, 2022).
[13] CodeAcademy.com. “Interview Prep.” https://www.codecademy.com/catalog/subject/interview-prep/. (Retrieved August 18, 2022).
[14] S. Cooper, W. Dann, and R. Pausch. 2003. “Teaching objects-first in introductory computer science.” *ACM SIGCSE Bulletin.,* 35, 1, 2003, pp. 191–195.
[15] M. E. Crosby and J. Stelovsky. “How do we read algorithms? A case study.” *Computer*. 23, 1, 1990, pp. 25–35.
[16] E. Dillon, B. Williams, A. Ajayi, Z. Bright, Q. Kimble-Brown, C. Rogers, M. Lewis, J. Esema, B. Clinkscale, and K.L. Williams. “Exposing Early CS Majors to Coding Interview Practices: An HBCU Case Study.” *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, IEEE, 2021, pp. 1-4.
[17] E. Dillon, and K. L. Williams. "Course content as a tool of inclusivity for Black/African-American women in computing." *Journal of Computing Sciences in Colleges*, 36(3), 2020. pp. 151-160.
[18] Z. Dodds, C. Alvarado, G. Kuenning, and R. Libeskind-Hadas. “Breadth-first CS 1 for scientists.” *ACM SIGCSE Bulletin*., 39, 3, 2007, pp. 23–27.
[19] EEOC.gov. “US Equal Employment Opportunity Commission. Diversity in High Tech.” https://www.eeoc.gov/special-report/diversity-high-tech/. (Retrieved August 18, 2022).
[20] D. Ford, T. Barik, L. Rand-Pickett, and C. Parnin. “The techtalk balance: what technical interviewers expect from technical candidates.” *In 2017 IEEE/ACM 10th International*

*Workshop on Cooperative and Human Aspects of Software Engineering (CHASE).,* 2017, pp. 43–48.
[21] M. H. Goldwasser and D. Letscher. “Teaching an object-oriented CS1- with Python.” *ACM SIGCSE Bulletin.,* 40, 3, 2008, pp. 42–46.
[22] K. Gosha, V. Kannan, L. Morgan, and E. W. Huff Jr. “Strategic Partnerships to Enhance Data Structures and Algorithms Instruction at HBCUs.” *In Proceedings of the 2019 ACM Southeast Conference.* 2019, pp. 194–197.
[23] P. Hall Jr and K. Gosha. “The effects of anxiety and preparation on performance in technical interviews for hbcu computer science majors.” *In Proceedings of the 2018 ACM SIGMIS Conference on Computers and People Research.* 2018, pp. 64–69.
[24] S. Horwitz, S. H Rodger, M. Biggers, D.Binkley, C. K. Frantz, D. Gundermann, S. Hambrusch, S. Huss-Lederman, E. Munson, B. Ryder, et al. “Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science.” *ACM SIGCSE Bulletin.,* 41, 1, 2009, pp. 163–167.
[25] A. Kapoor and C. Gardner-McCune. “Introducing a Technical Interview Preparation Activity in a Data Structures and Algorithms Course.” *In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education.*, 2, 2021, pp. 633–634.
[26] KARAT.com. “The Karat interview process.” https://karat.com/candidates/how-it-works/. (Retrieved August 18, 2022).
[27] M. Lev-Ram. “Apple commits more than $50 million to diversity efforts.” http://fortune.com/2015/03/10/apple-50-milliondiversity/. (Retrieved August 18, 2022).
[28] Bureau of Labor Statistics. “Occupational Employment Projections to 2022, Bureau of Labor Statistics Monthly Labor Review.” https://www.bls.gov/opub/mlr/2013/article/occupationalemployment projections-to-2022.htm/. (Retrieved August 18, 2022).
[29] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. Paterson. “A survey of literature on the teaching of introductory programming.” *Working group reports on ITiCSE on Innovation and technology in computer science education*., 2007, pp. 204–223.
[30] M. Simon. “Google Sets Sights on HBCUs for Recruitment Efforts.” https://www.nbcnews.com/news/nbcblk/googlesets-sights-hbcus-recruitment-efforts-n601476/. (Retrieved August 18, 2022).
[31] UCI. “CSIC: The Importance Of Computer Science Education.” https://sites.uci.edu/cs1c/importance-of-computer-scienceeducation/. (Retrieved August 18, 2022).
[32] Udemy.com. “Master the Coding Interview. Data Structures + Algorithms.” https://www.udemy.com/course/master-thecoding-interview-data-structures-algorithms/. (Retrieved August 18, 2022).
[33] A. N. Washington, L. Burge, M. Mejias, K.Jean-Pierre, and Q. Knox. “Improving undergraduate student performance in computer science at historically black colleges and universities (HBCUs) through industry partnerships.” *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education.,* 2015, pp. 203–206.
[34] S. Wiedenbeck, V. Ramalingam, S. Sarasamma, and C. L. Corritore. “A comparison of the comprehension of object-oriented and procedural programs by novice programmers.” *Interacting with Computers.,* 11, 3, 1999, pp. 255–282.

[35] Yale.edu. “Yale University: Office of Career Strategy. The Technical Interview.” https://ocs.yale.edu/channels/the-technicalinterview/. (Retrieved August 18, 2022).