Exposing Early CS Majors to Coding Interview Practices: An HBCU Case Study*

Edward Dillon*, Briana Williams, Ayomide Ajayi, Zipporah Bright, Quinlan Kimble-Brown
Chauncey Rogers, Myles Lewis, Joseph Esema, Ben Clinkscale
Computer Science Department
Morgan State University
Baltimore, USA
{edward.dillon*, briwil95, ayaja3, zibri1, qukim1, chrog8, mylew3, joese1, becli1}@morgan.edu

Krystal L. Williams
Educational Leadership, Policy, Technology Studies Department
The University of Alabama
Tuscaloosa, USA
krystal.l.williams@ua.edu

***Abstract*— The demand for candidates to fill job vacancies in the field of Computer Science (CS), especially in the industry sector, is high. Yet, the supply of CS candidates to fill these positions remain low. Student attrition in CS departments and their inability to produce a sufficient representation of candidates can attribute to one aspect of this problem. Another reason is a candidate's inability to showcase the appropriate skill-sets and related preparation needed for such career-based opportunities.**

One approach has been to incorporate industry-based practices into the CS classroom as a way to equip and prepare majors for such expectations. Coding interview exercises are one notable practice. Aspects of this particular practice have been examined in academic settings. However, there is a lack of current studies that examine this type of practice at earlier stages of a CS curriculum.

This article discusses a case study for exposing early CS majors to relative coding interview practices in the form of whiteboard problem solving. During the Fall 2020 semester, a PRE and POST virtual whiteboard problem-solving exercise were conducted on a CS2 and Object-Oriented Programming course at a Mid-Atlantic HBCU in the United States. The results revealed that majority of the students in both courses were able to complete the problem sets for both exercises successfully. Likewise, both groups expressed a favorable perception about these exercises, and exhibited adequate levels of comfort for completing these exercises. However, both groups also showed adequate levels of anxiety.

Keywords*— *Early CS Majors, Whiteboard Problem Solving, Coding Interviews, Black/African-American students

I. Introduction

Due to the continuing evolution of technology, jobs and career-based opportunities are ever increasing, thus making the supply of candidates to fill these positions more challenging. Computer Science (CS) has been a growing and increasingly lucrative field for some time. The demand of job applicants in this field are high but the supply of CS candidates to fill these positions are relatively low. During 2020, there were 400,000 college graduates looking for jobs in the industry while there were nearly 1.4 million 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 on last year [11]. This imposes the question for how can a sufficient supply of candidates be produced to meet this demand? Moreover, this poses an additional burden on CS and related computer-based departments to increase their productivity of graduates who can fill these positions as practitioners. It is known that CS departments tend to struggle with such productivity due to the high attrition rates of students leaving the major [1]. Another challenge that technology fields like CS face is producing quality graduates who possess strong skill-sets and sound computational proficiencies to meet the qualifications and expectations imposed by many of these job solicitations.

This introduces another of question of how can CS and departments alike strengthen their ability to effectively prepare their students for such expectations? When CS majors are enrolled in computation programming courses, they are expected to grasp necessary programming concepts and paradigms for success. Moreover, these students must become adept with employing the appropriate syntax and semantics from the programming language being used in these courses, regardless of language simplicity or complexity. Thirdly, they must exhibit some level of proficiency for effectively operating the programming tool or editor assigned for those courses. Yet, it is possible for students to be unable to showcase these developed skill-sets in a way that impresses upon an employer that they are a worthy candidate.

One practice that has been used by industry-based tech companies to screen the potential of a candidate's computational skills is *coding interviews*. This practice not only involves solving technical problems during an interview but also requires job applicants to vocalize their thought process when solving this problem. Even though understanding a problem and writing code are critical in the coding interview process, being able to show one's ability to solve computational problems effectively is of paramount importance. Moreover, it gauges a candidates' level of confidence during this interaction, while also serving as an indicator for screening whether a candidate possesses the necessary technical and verbal interpersonal skills needed to be effective for the intended job position of pursuit [5].

Based on the breadth of skills (both *technical* and *interpersonal*) that students are expected to exhibit during a professional coding interview session, exposing such practices to students as they matriculate through a CS curriculum may be warranted. To address such a possibility, our study proceeds to investigate the impact of exposing computer science majors, who are early in the CS curriculum, to coding interview practices in the form of whiteboard problem solving. Through gauging the students' confidence levels on whiteboard problem solving, an educational practice that mirrors technical interviews conducted by a variety of technology companies, we can examine the mentality of these students as they are exposed to such practices. After targeting the particular discrepancies that students are exhibiting, it will be critical to rectify these issues in order to help the students improve their skill-sets, competencies, and proficiencies with computational problem-solving.

II. LITERATURE REVIEW

Whiteboard interaction is a common practice that can be seen in many academic classrooms. When emphasizing the critical importance of problem solving in STEM subjects like CS and Mathematics, the use of whiteboards for this style of learning has been applicable to promote student engagement [4], and representation skills of their acquired knowledge [8]. The latter aspect is a key attribute that tech companies use to evaluate a candidate during a technical or coding interview. One reason is that this style of evaluation showcases a candidate's *mental* and *cognitive* state, and *critical thinking* processes as it relates to their acquired knowledge [2].

When emphasizing current literature as it relates to coding interviews, there have been recent studies conducted to evaluate the impact of such practices. Ford et al. [5], for instance, conducted a comprehensive review analysis on 70 interview evaluations that stemmed from 9 different tech companies. The objective was to gauge notable trends and consistent attributes that employers look for in candidates during a coding interview session. From this work, it was discovered that companies can respectively have different attributes for how a candidate is evaluated. Likewise, the candidates' ability to engage in open dialogue and to seek clarity by asking their own set of questions as it relates to an asked question from an employer/interviewer is expected.

Hall and Gosha [7] conducted an empirical study that surveyed junior and senior African-American CS majors, who were enrolled at an historically black college and university (HBCU). This study captured information from these students on the basis of personal experiences with coding interviews, plans of preparation used to practice for these interviews, and whether *anxiety* plays an integral role during their participation in coding interviews. From this work, it was found that students' prior experience with coding interviews actually assisted in reducing their levels of anxiety when participating in later interviews. It was also argued that anxiety is an underlying factor that could determine a student's overall performance in an interview. It was concluded from this study that as students become more exposed to coding interview practices their anxiety decreases, while in turn their overall performance increases.

Even though literature notes prior efforts for studying the impact of coding interviews amongst CS majors, more systematic and empirical evaluations of such practices are needed. When it comes to the representation of Black/AA practitioners in industry, more efforts as seen by Hall and Gosha [7] are also warranted. To emphasize the HBCU dynamic and their specific role in increasing Black/AA representation of CS practitioners, there have been initiatives by tech companies, who are making efforts to address the retention pipeline of underrepresented minorities at these specific institutions [9-10]. Likewise, these companies have begun working closely with minority-serving institutions in an effort to provide insight on the type of computational skills and programming proficiency a student (or a future prospective employee) must exhibit in order to be successful in these particular sectors [3, 6, 12]. One common insight from their observations concerns a candidate's ability to exhibit proficient critical thinking skills to solve problems through computational programming. Based on its noted benefits, being able to effectively showcase their technical and interpersonal skills during coding interview sessions could be one practice that can assist students in meeting such skill expectations.

To aid in expanding the contributions of these current efforts, our work provides an empirical-based systematic study involving coding interview practices that was conducted on Black/AA CS majors enrolled at an HBCU located in the Mid-Atlantic United States. In contrast to Hall and Gosha's work, our study looks at the impact of such practices on students who are even earlier in the CS pipeline.

III. METHODS

The objective of the Whiteboard Problem Solving study was to examine the students' ability to conduct critical thinking, verbally communicate their ideas, and create solutions to a given problem. This study was comprised of a *PRE* and *POST* assessment to gauge their problem solving abilities at different points throughout the semester. The targeted participants for this study were students enrolled in either the CS2 or Object-Oriented Programming (OOP) course at this particular HBCU. CS2 students are traditionally taught intermediate programming concepts and structures using Python programming. The demographic make up for CS2 are primarily *freshmen*, and occasionally *sophomore* CS majors. OOP students are traditionally taught advanced programming and object-oriented structures using C++ programming. The demographic makeup for OOP are *advanced sophomores* and *junior* CS majors. For both courses, the ethnic representation of these students were predominantly Black/AA.

Since this study occurred during the Fall 2020 semester, adjustments had to be made to account for these courses being taught remotely due to the COVID-19 pandemic. Rather than using physical whiteboards as initially planned for this study, the Zoom Video/Web Conferencing system was used as a virtual alternative (Fig. 1).

Fig. 1. Zoom Video/Web Conferencing System

Funding Agency: NSF grant nos. *HRD-1912098* and *HRD-2011793*.

For both the *PRE* and *POST* assessments, there were two tasks assigned. During the *PRE* assessment, the students 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 a breakout room to solve this problem using an programming tool or editor of their choice. As part of this process, each group had to collectively explain aloud their critical thinking and reasoning for approaching this problem, and were encouraged to generate pseudocode that reflected this reasoning. Afterwards, they would officially implement their solution using their editor and given programming language syntax (*Python* or *C++)* to verify that their solution is *correct*. Each group had 30 minutes to complete the first task. Afterwards, all groups returned to the main room to be randomly assigned a second procedural programming problem from a list of five problems and return to their breakout rooms to repeat the same procedure for another 30 minutes. Upon completing the *Task 2*, the students were given a survey to complete regarding this experience. Finally, each group submitted their recordings as a *.mp4* file. The protocol for the *POST* assessment was primarily similar to the *PRE* assessment. The only difference was that during *Task 2* the students in both courses, respectively, were randomly assigned an OOP problem from a list of four to solve in their assigned groups.

IV. Research Questions

There were two primary research questions proposed for this study. Corresponding findings for these questions are addressed in the Section VI (Results & Findings).

RQ1: *Did prior experience with coding interviews influence how students performed on these whiteboard problem-solving assessments?*

RQ2: *Were there favorable changes in the students' perception, comfort, and anxiety levels when matriculating from the PRE to the POST assessment?*

V. Empirical Analysis

The analysis discussed in this section focuses on the feedback given by the students on the survey component of the *PRE* and *POST* assessments, respectively. The key attributes that are discussed are *prior coding interview experiences, task completion, perception, comfort levels,* and *anxiety levels.* Tables I-V provides detailed descriptions of each attribute.

TABLE I. PRIOR CODING INTERVIEW EXPERIENCE

Prior Coding Interview Experience Percentages		
	CS2 (N=13)	**OOP (N=20)**
Coding Interview Type	*%*	*%*
Face-to-Face	Yes: 31% No: 69%	Yes: 37% No: 63%
Virtual/Phone-Based	Yes: 23% No: 77%	Yes: 26% No: 74%

TABLE II. TASK COMPLETION – DESCRIPTIVE DATA

Task Completion Percentages				
	CS2		**OOP**	
Assessment Type	**N**	*%*	**N**	*%*
PRE Assessment (Task #1) *Procedural/Functional Programming*	13	Yes: 92% No: 8%	20	Yes: 95% No: 5%
PRE Assessment (Task #2) *Procedural/Functional Programming*	13	Yes: 85% No: 15%	20	Yes: 75% No: 25%
POST Assessment (Task #1) *Procedural/Functional Programming*	11	Yes: 82% No: 18%	15	Yes: 80% No: 20%
POST Assessment (Task #2) *OOP Programming*	11	Yes: 55% No: 45%	14*	Yes: 86% No: 14%
**one student did not provide an answer*				

TABLE III. PERCEPTION: INITIAL & IMPROVED CHANGE

Perception: Initial (PRE) & Changes (POST)				
	CS2		**OOP**	
Assessment Type	**N**	*%*	**N**	*%*
Initial Perception	12*	FR: 58% N: 17% UR: 25%	20	FR: 75% N: 15% UR: 10%
Improved Change in Perception	11	Yes: 82% No: 18%	15	Yes: 60% No: 40%
FR = *Favorable Response* N = *Neutral* UR = *Unfavorable Response* **one student did not provide an answer*				

TABLE IV. COMFORT LEVELS: BEFORE & AFTER ASSESSMENTS

Comfort Levels						
	CS2			**OOP**		
Assessment	**N**	**Mean B/A**	***% Increase***	**N**	**Mean B/A**	***% Increase***
PRE	12*	71/79	10.88	20	71/85	19.01
POST	11	69/70	2.65	15	77/84	8.53
B/A =*Before/After (Assessment)* **one student did not provide an answer*						

TABLE V. ANXIETY LEVELS

Anxiety Levels				
	CS2		**OOP**	
Assessment	**N**	**Mean**	**N**	**Mean**
PRE	13	43	20	51
POST	11	37	15	51
% Increase		-13.95%		0.00%

In Table I, the percentages revealed that the majority of the students in CS2 and OOP, respectively, did not possess any prior coding interview experiences before these completing these assessments. In spite of lacking prior experience, the percentages in Table II revealed that majority of the students in both courses, respectively, completed their assigned whiteboard problem solving tasks. This was found to be true during both the *PRE* and *POST* assessments, even though the CS2 students showed a slightly higher percentage for not being able to solve their assigned OOP problem during the *POST* assessment. Likewise, the percentages in Table III revealed that majority of these students in both courses, respectively, exhibited an initial perception that was favorable towards problem solving in this format during the *PRE* assessment. Moreover, this perception was found to improve after completing the *POST* assessment. Regarding their comfort levels, these responses were based on 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. For each assessment, the students showed an adequate level of comfort upon their initial exposure to these assessments, which also increased after completing a given assessment (Table IV). When it came to their anxiety levels, these responses were also based on a 10-point Likert scale (1=*not anxious at all*, 10=*absolutely anxious*). The scores were then normalized on a scale of 0 to 100, where the lower the score the higher the anxiety. The CS2 students exhibited adequate levels of anxiety while also showing an increase in their anxiety levels as they matriculated from the PRE to the POST assessment. The OOP students also exhibited an adequate level of anxiety during these assessments even though these levels remained steady during both assessments (Table V).

VI. RESULTS & FINDINGS

When revisiting both research questions posed in Section IV, it can be determined from these results that having prior coding interview experiences did not play an integral role in influencing the students' performance on these whiteboard problem solving assessments. Majority of these students were able to complete the given tasks even though many of them did not have prior coding interview experience. Likewise, they exhibited favorable perceptions towards this style of problem solving as well as adequate comfort levels, which increased after completing a given assessment. However, their anxiety levels were also found to be consistently adequate throughout these assessments.

VII. CONCLUSION & FUTURE WORK

The purpose of this article was to examine the impact of exposing early CS majors who were primarily Black/AA to coding interview practices using whiteboard problem solving. It was found that this exposure garnered favorable psychosocial behaviors from the students even though many of them lacked prior experiences with the coding interview style of problem solving. As seen in Hall and Gosha's work [6], the students' also exhibited adequate levels of anxiety when exposed to these practices.

One future work is to further gauge the underlying factors that could cause these anxiety levels in students. Another future work is to conduct empirical-based comparisons between multiple cohorts of students as they matriculate through a CS curriculum. A third future work will be to re-assess students who were once in the CS2 and/or OOP courses later in the CS pipeline to measure the longitudinal impacts of being exposed to this style of problem solving.

ACKNOWLEDGMENT

This current work is funded by the National Science Foundation under grant nos. *HRD-1912098* and *HRD-2011793*.

REFERENCES

[1] Beaubouef, T. and Mason, J. 2005. Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. Inroads 37(2), 2005, pp. 103-106.

[2] Behroozi, M., Lui, A., Moore, I., Ford, D. and Parnin, C., 2018, May. 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* (pp. 93-96).

[3] Cain, C. C., Buskey, C., Bryant, A. M., Washington, G., & Burge, L. Research Implications of the Tech Exchange: Immersion of Howard University Computer Science and Informatics Students in Silicon Valley, 2019.

[4] Cassapu, A., 2009, July. Introducing the interactive whiteboard in computer science teaching: a high school experience. In *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education* (pp. 364-364).

[5] Ford, D., Barik, T., Rand-Pickett, L., & Parnin, C. (2017, May). The tech-talk balance: what technical interviewers expect from technical candidates. In *2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (pp. 43-48). IEEE.

[6] Gosha, K., Kannan, V., Morgan, L., & Huff Jr, E. W. Strategic Partnerships to Enhance Data Structures and Algorithms Instruction at HBCUs. In *Proceedings of the 2019 ACM Southeast Conference*, pp. 194-197, April 2019.

[7] Hall Jr P, Gosha K. The Effects of Anxiety and Preparation on Performance in Technical Interviews for HBCU Computer Science Majors. InProceedings of the 2018 ACM SIGMIS Conference on Computers and People Research 2018 Jun 18 (pp. 64-69).

[8] Hwang, W.Y., Chen, N.S., Dung, J.J. and Yang, Y.L., 2007. Multiple representation skills and creativity effects on mathematical problem solving using a multimedia whiteboard system. *Journal of Educational Technology & Society*, *10*(2), pp.191-212.

[9] Lev-Ram, M. (2015, March 10). Apple commits more than $50 million to diversity efforts [Online Article]. Retrieved November 26, 2018, from http://fortune.com/2015/03/10/apple-50-million-diversity/, March 2015.

[10] Simon, M. Google Sets Sights on HBCUs for Recruitment Efforts [Online Article]. Retrieved November 26, 2018, from https://www.nbcnews.com/news/nbcblk/google-sets-sights-hbcus-recruitment-efforts-n601476, June 2016.

[11] The Importance Of Computer Science Education. Retrieved March 1, 2021, from https://sites.uci.edu/cs1c/importance-of-computer-science-education/.

[12] Washington, A. N., Burge, L., Mejias, M., Jean-Pierre, K., & Knox, Q. A. 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*, pp. 203-206, February 2015.