The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #41778

# Educational Expertise: Faculty Insights on Preparing Computing Students to Navigate Technical Interviews

**Stephanie Jill Lunn, Florida International University**

Stephanie Lunn is an Assistant Professor in the School of Universal Computing, Construction, and Engineering Education (SUCCEED) and the STEM Transformation Institute at Florida International University (FIU). She also has a secondary appointment in the Knight Foundation School of Computing and Information Sciences (KFSCIS). Previously, Dr. Lunn earned her doctorate in computer science from the KFSCIS at FIU, with a focus on computing education. She also holds B.S. and M.S. degrees in computer science from FIU and B.S. and M.S. degrees in neuroscience from the University of Miami. In addition, she served as a postdoctoral fellow in the Wallace H. Coulter Department of Biomedical Engineering at the Georgia Institute of Technology, with a focus on engineering education. Her research interests span the fields of computing and engineering education, human-computer interaction, data science, and machine learning.

**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 &gt;$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.

**Zubayer Ahmed Sadid, Florida International University**

# Educational Expertise: Faculty Insights on Preparing Computing Students to Navigate Technical Interviews

## Abstract

Obtaining a computing position can entail hiring practices distinct from other domains in that they frequently involve technical interviews, an approach which emphasizes real-time evaluation of programming abilities. Apart from requiring that job candidates find the correct solution, they are also encouraged to find one that is efficient and optimal while also speaking through their thinking. Although computing students may be familiar with the theoretical and foundational topics necessary to succeed, traversing the employment process may be particularly daunting and necessitate immense preparation. To better understand how education may enhance students' readiness, we conducted a focus group with ($n$ = 7) faculty members from around the United States. We applied social cognitive career theory as we explored: 1) learning experiences that educators may incorporate into courses; and 2) actions perceived as necessary for institutions to enhance students' technical interview performance and career attainment. We employed thematic analysis to assess their responses and suggestions on ways forward. Within courses, three themes emerged, as educators described opportunities for "fostering professional skills" (e.g., communication and teamwork) and "fostering technical skills" (e.g., use of tools, programming problem decomposition, and testing). They also spoke about job preparation, including utilizing role play in the context of mock interviews and the value of "application material development" (e.g., cover letters, resumes). When considering the broader institutional actions, two themes emerged: "supportive practices" and "technical interview and career knowledge." Participants touched on the possibility of faculty receiving training themselves, since often they stayed in academia without applying for an industry position and were unfamiliar with expectations to advise students on what to anticipate. They also highlighted the value of industry partnerships and student organizations for hiring preparation and networking. Meanwhile, several suggested that existing services, including platforms and/or groups that were already offering mock interviews or training (e.g., Big Interview, Brilliant Black Minds) could provide students with additional awareness and practice to feel more comfortable with what to expect. It is our goal that the findings and recommendations made in this paper encourage other faculty to recognize how they could further aid in students' preparation and integrate it into lessons. Beyond that, we hope that educators and administrators consider possible ways to aid in computing students' understanding of technical interviews and seek to enhance their graduate employability.

## 1 Introduction

Computer and information technology occupations are expected to explode over the next decade, adding an estimated 377,500 jobs annually, according to the United States (U.S.) Bureau of Labor

Statistics [1]. In particular, for Software Developers, Quality Assurance Analysts, and Testers, the average growth rate is expected to surge by 25% (much higher than the 3% for all occupations). For Computer and Information Research Scientists, the growth rate is projected to be 23%. Despite clear needs to fill such roles, obtaining a computing job can be challenging, something often attributed to, at least partially, the expectations of the hiring process and graduates' performance on technical interviews [2].

**Technical interviews**, also referred to as coding interviews or programming interviews, are a component of the hiring process frequently employed for computing roles [3]. They entail asking job candidates to solve problems on a whiteboard or using a text editor to evaluate their knowledge, skills (e.g., communication), capabilities (e.g., technical and performance), dispositions (e.g., adaptability), and thought processes. Yet, while students may be taught computing foundations and theory throughout their education, this does not always translate into positive outcomes. According to a recent evaluation of performance at interviewing.io, only 54% of candidates actually pass technical interviews [4].

Although such approaches may be commonplace to evaluate candidates for computing roles, they are often criticized. An exploration of HackerNews, a social news website for those involved in software development, has previously described how they can not only induce anxiety, as may be more common in any interview situation, but also in how they evaluate candidates' thinking [5]. They are seen as unrealistic for what coding may look like in practice. Furthermore, the expectations around being able to solve problems in an efficient and optimal fashion as quickly as possible necessitate significant preparation and memorization of algorithms.

Students are encouraged to begin preparing months or even years in advance to succeed in technical interviews [6, 3]. Such preparation may include mock interviews, the use of online tools (e.g., LeetCode), or textbook aids (e.g., Cracking the Code Interview Book [3]) [7]. However, practically, students have other commitments that may limit their availability to do so [8]. Instead, students have reported that they may actually have begun preparing for the technical interview(s) 1 week or less before their interview [8].

While ideally the industry would find alternative approaches to assessing candidates, current hiring practices are so widespread that they are unlikely to be changed anytime soon. So what can be done to help students excel in technical interviews and aid in their transition to the workforce? How can higher education institutions foster the knowledge, capabilities, skills, and dispositions required for students to succeed in the workplace and enhance their employability?

In this study, we sought to explore the opportunities to integrate such awareness and training into curricula. To better understand where it may be feasible to do so within existing academic and programmatic structures, we elected to focus on understanding through postsecondary educators. The research questions (RQs) that we sought to answer were:

- **RQ1:** *How do educators incorporate learning experiences for undergraduate computing students into courses to aid in their career outcome expectations?*
- **RQ2:** *What actions do educators see as necessary for institutions to take to enhance students' technical interview performance and career attainment?*

Towards our goal, we conducted a focus group with educators from institutions around the U.S.

We centered deliberately on those who may interact with and/or teach computing students, a term that we envision as encompassing majors such as computer science (CS), software engineering (SE), information technology (IT), computer engineering (CE), or data science (DS). We want to acknowledge that there may be other majors where students ultimately plan to pursue computing roles and thus will need to complete technical interviews, given the expanding subfields of computing and the interdisciplinarity of majors. However, this work was intended as a pilot to provide initial insight, and such expansion was beyond the scope of the current investigation.

In the paper that follows, we first present the background important for this investigation in Section 2. We then describe the theoretical framework that guided this inquiry in Section 3. In Section 4, we elaborate on our methods, describing the participants, the focus group, and thematic analysis. The results of our study and a discussion of our findings are described in Sections 5 and 6. In Section 7, we address the limitations of our study. Finally, we present the key takeaways and their implications for other researchers and/or educators in Section 8.

## 2 Background

In the sections below, we first differentiate what is meant by knowledge, skills, capabilities, and dispositions. Such expansion is important for framing what may be investigated during the hiring process. We then elaborate on how preparation for technical interviews in academia may manifest.

### *2.1 Computing Knowledge, Skills, Capabilities, and Dispositions*

**Knowledge** refers to the facts or information that may be obtained through an individual's experience or education. For the purpose of our investigation, we focus on computing knowledge. Although not the only knowledge required, a key component of computing is learning to program, which also necessitates being able to find solutions as well as reading and tracing code [9]. It can also entail more specific forms of knowledge, like being able to demonstrate syntactic knowledge [10].

Comparatively, **skills** embody the knowledge acquired and can manifest as performing a particular task [11]. These are considered things that can be taught, developed, or learned over time. However, an individual can also be **capable** of doing something, a term that refers to having the potential and possessing the skills to accomplish a particular task. Here, we consider skills in reference to technical skills and professional skills that may be needed for long-term success in computing [12].

Please note that professional skills may also be referred to as "social skills" or "soft skills." However, it has been suggested that the term "soft" can imply that they are less relevant than "hard" or technical skills [13]. Accordingly, we use "professional skills" in the work that follows and apply the definition provided by others for computing in that we recognize it as "a non-technical skill that can be learned in a traditional academic setting and is relevant to proficiency in a given field of study, which may itself be largely technical" [14, p. 288].

Finally, **dispositions** are described in terms of the way an individual may apply skills or knowledge to impact their perspectives or values. Scholars have previously distinguished between these by describing knowledge as the "know what," the professional skills as the "know how," and dispositions as the “know why” [15]. In a professional context, dispositions have been described as vital to task completion [16], and can be critical in the workplace [17, 18]. Therefore, we also considered how they may appear in computing students.

### *2.2 Preparation for Technical Interviews in Academia*

Several studies have focused on exploring the experiences of students and developers as they have traversed the hiring process in computing and considered what preparation may entail [7, 5, 19, 20, 8, 21, 22, 23]. Among a group of undergraduates who completed one or more technical interviews, roughly a third of those surveyed reported feeling unprepared [21]. Often, students noted that they were made aware of what to expect and received most of their guidance and preparation for technical interviews from their social network and organizations/clubs on campuses [23]. Given that students were often limited in availability due to other commitments (e.g., caring for a family or working in another job), they mentioned that they would like to have more training incorporated into their education [22]. Preferably, they envisioned the best option being a stand-alone course dedicated to problem solving in the manner expected during technical interviews as part of the curriculum.

Although not presently widespread, there have been several efforts towards exploring training opportunities in academia. One of the few stand-alone courses that may be available is through Stanford, a one-credit course called “CS 9: Problem-Solving for the CS Technical Interview”[1]. Several other scholars have described additional ways they have been able to integrate such preparation into existing classes.

Previously, Kapoor and Gardner-McCune have worked to include technical interview awareness and problem-solving opportunities in a Data Structures and Algorithms course [24]. They created a panel of undergraduate teaching assistants who had worked at top technology companies and who could share their experiences with preparation and what the hiring process entailed. Later in the semester, they also included mock interviews. Feedback from the students demonstrated they found the activities beneficial, not only serving to inform but also to motivate them to apply for internships and jobs and mitigating the anxiety they felt about the process.

Dillon et al. [25] also previously implemented interactive whiteboard problem-solving into CS2 and Object-Oriented Programming (OOP) courses. Using a collaborative approach, students were assigned small groups via Zoom and tasked with working together to find solutions. Pre- and post-experience assessments demonstrated that this approach helped them to conceptualize and tackle problems, as well as being able to better articulate their solution throughout, as expected during technical interviews.

While the examples shared above represent an exciting shift that could help students achieve knowledge transfer of concepts and practically enhance their preparation for the hiring process, more is needed. Also, less is known about the familiarity of the hiring process in computing from the perspective of educators. Likewise, it is unclear how or if they envision technical interview training and preparation fitting into academic curricula. Accordingly, we sought to answer these questions in the investigation performed.

[1] https://web.stanford.edu/class/cs9/

## 3 Theoretical Framework

We employed Social Cognitive Career Theory (SCCT), a framework focused on career development [26]. It not only recognizes the individual factors that may contribute to academic and vocational interests and attainment, but also the environmental and behavioral influences [27]. Based on Bandura's self-efficacy [28], it considers the factors that may have an impact on and determine performance, including interactions with others and personal achievements, and how they may contribute. Ultimately, this leads to outcome experiences — the perceived results of taking certain actions [26]. It seeks to describe how interventions and activities can enhance personal mastery experiences. Furthermore, in the context of computing, it has been shown to describe not only interests, but also choice goals [27].

An overview of the framework, as it pertains to our study, is presented in Figure 1. We consider learning experiences in terms of professional and technical skill development as they may pertain to foundations in the field and the hiring process. We also consider these variables as they ultimately lead to performance domains and attainments in the form of technical interviews and computing jobs.

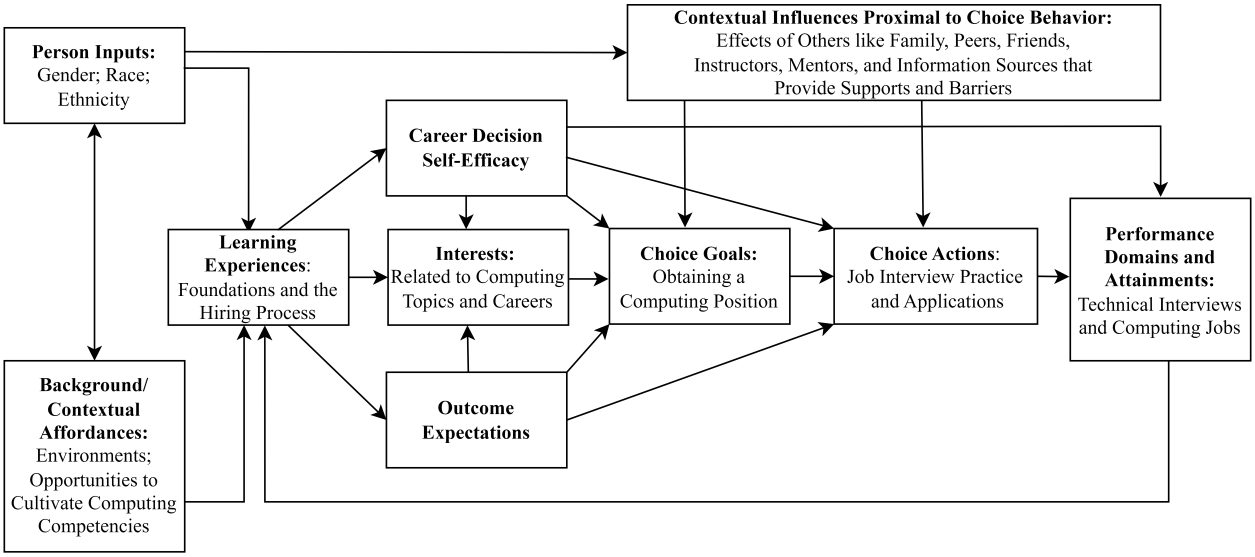
Person Inputs: Gender; Race; Ethnicity
Contextual Influences Proximal to Choice Behavior: Effects of Others like Family, Peers, Friends, Instructors, Mentors, and Information Sources that Provide Supports and Barriers
Career Decision Self-Efficacy
Learning Experiences: Foundations and the Hiring Process
Interests: Related to Computing Topics and Careers
Choice Goals: Obtaining a Computing Position
Choice Actions: Job Interview Practice and Applications
Performance Domains and Attainments: Technical Interviews and Computing Jobs
Outcome Expectations
Background/ Contextual Affordances: Environments; Opportunities to Cultivate Computing Competencies

Figure 1: SCCT framework, adapted from [26]

We applied SCCT in our inquiry to better explore how learning experiences could impact students' self-efficacy, outcome expectations, interests, goals, choice actions, and performance domains and attainments. Specifically, we explored these relationships as perceived by educators to learn more about what they observed based on the situations within their respective courses and departments. Our choice was intended to span not just the diversity of the departments, but also the students, institutions, and resources offered. It was applied in our study design, informing the RQs and focus group prompts, and in the interpretation of our findings.

## 4 Methods

A qualitative study was conducted to understand how educators may view the hiring process in computing and their suggestions around how preparation could fit within existing institutional structures. We elected to do so through a focus group held with educators from around the U.S.

who had agreed to participate in a multi-session technical interview workshop. Institutional Review Board approval was obtained prior to beginning the study.

The choice was deliberate to minimize their familiarity with the process which could impact responses.

### *4.1 Participants*

As part of an initial pilot, $n = 7$ educators participating in the workshop were enlisted for the focus group. These participants elected to join voluntarily and were recruited from institutions throughout the U.S. through various organizational mailing lists. Among those who joined, they came from a mix of institutional types, including 2- and 4-year colleges, with varying levels of research activity and student demographics.

### *4.2 Data Collection*

Although the workshop was held for a fixed duration over multiple weeks throughout the academic year, the focus group was conducted during the first session, which was hosted over Zoom. The focus group portion lasted approximately one hour. The moderator used the script shown in the Appendix, and the session was audio recorded after receiving the consent of the participants.

### *4.3 Data Analysis*

Given that the goal was to understand the educators' perspectives and experiences within and across their institutions, we elected to use thematic analysis [29]. We followed the procedure previously described [30], from familiarization with the data; to developing smaller codes; to generating, reviewing, defining, and naming the overarching themes related to answering the RQs. All the audio collected from the focus group was saved and transcribed by a third-party transcription service. The transcripts were then manually confirmed.

Upon completion, the first and third authors independently reviewed the transcript. They iteratively read and developed a codebook for each RQ, identifying and organizing key components present into codes, with definitions, and broader themes. They then met to negotiate and finalize each codebook. Upon completion, these authors used the codebook to separately code the transcript in NVivo. We obtained a pooled Cohen's Kappa of 0.78 for the first codebook and 0.72 for the second, both considered a "substantial agreement," as defined according to the range between 0.61 and 0.80 [31].

### *4.4 Positionality*

The first author is an assistant professor of computing and engineering education and researcher who has previously examined students' experiences with the hiring process in computing. She has undergone technical interviews and received job offers in the field, but has not worked in a computing role in industry. She was responsible for study design, leading the focus group, co-leading the analysis, and the preparation of the manuscript. She leveraged her familiarity of technical interviews and what they may look like, as well as her teaching experience, into the questions asked in the focus group. Her background may have also allowed her to form a rapport with the other educators in her role as moderator.

The second author is an associate professor in computer science. He has also spent a portion of a summer at Google's main campus in Mountain View, CA as a Faculty in Residence. Key reasons for this visit was to: 1) gain insight into industry expectations and engineering/developer practices in this type of culture, and 2) develop effective computational and learning strategies to be employed in his classroom to help CS majors develop proper skill-sets that can be applicable and transferable throughout their careers. This experience is one of the catalysts for this author's current work and interest in technical interview prep research in academia. For this manuscript, this author assisted in the study design, led the recruitment of the focus group participants, and assisted in the manuscript's preparation.

The third author is currently an undergraduate student pursuing a bachelor's degree in computer science. This author has never undergone a technical interview but is currently in the process of searching for computing internships. He contributed to the thematic analysis for this study.

## 5 Results

Below we present the results of the thematic analysis in relation to each RQ.

### *5.1 RQ1: How do educators incorporate learning experiences for undergraduate computing students into courses to aid in their career outcome expectations?*

As shown in Figure 1, we observed three over-arching themes in the data: "Fostering Professional Skills"; "Fostering Technical Skills"; and "Job Preparation." These themes were compromised of seven codes. Below, we elaborate further on what each theme entailed.

Table 1: RQ1: Resulting themes and codes

| **Theme** | **Code** | **Description** |
|---|---|---|
| *Fostering Professional Skills* | Teamwork | Being able to collaborate and/or effectively and efficiently work with others |
| | Communication | Being able to express oneself and their work clearly and successfully |
| *Fostering Technical Skills* | Use of Tools | Offering the opportunity to use existing tools or platforms to acquire understanding or practice |
| | Testing | Examining artifacts for issues |
| | Programming Problem Decomposition | Descriptions of helping students to break down coding problems bit by bit, strategies they could employ, or ways to think through the solutions when they get stuck |
| *Job Preparation* | Interview Role Play | Mock interview or running through interview scenarios with pretend interviewer and interviewees |
| | Application Material Development | Helping students apply for a particular role or developing career content (e.g., resumes, cover letters) |

The first theme, "Fostering Professional Skills," spoke to the ways educators sought to impart the value of and incorporate experiences to encourage teamwork and communication. They articulated how interpersonal dynamics can be vital to accomplishing goals. The educators also highlighted the benefits of negotiation and learning how to navigate group situations. While students might not always enjoy being part of teams, it was seen as critical to their development. As one educator mentioned:

> *It's the same for me, we do a lot of group work to get them used to the fact that, you know, they're going to work with other people. And everything is practical as much as possible. I hate lecturing. So I do that practical aspect. And sometimes I let them pick groups, but most often times, I just tell them you're working with these two other people. And that's that. Figure it out.*

Others spoke to the importance of peer support throughout struggles students may face in academia or the workplace. As a result, they saw team-building as a valuable way to strengthen a community. One instructor elaborated, "*I started to do that, like actually trying to get them to work together more, because I want, because there's so few, there's so few women, and there's so few, like you, African Americans, and Latinos.*"

Communication was cited as critical in the job interview process and more broadly. Several of the educators described the need for students to be able to clearly present ideas to a range of audiences and how practice could be beneficial. They found distinctive ways to make this part of their lessons. For example, as one participant mentioned:

> *I also have at least one presentation in every single course, no matter what. I need to get them to stand in front of people and feel comfortable in speaking to a group, to a person. I sometimes make them record themselves. So, I change things up depending on the context too, but I feel that loosens them up enough for them to be able to speak to another human and not feel awkward.*

The educators also spoke about "Fostering Technical Skills" and how they worked to cultivate them. They detailed utilizing different activities in their classroom, tools, and ways they may have helped to enhance engagement and understanding, from Quizlets to Kahoot! Various platforms and online integrated development environments could also make it easier for students to learn and practice fundamental concepts. As one participant described:

> *Replit, I use that for my non-majors, who I teach Python to and also for my intro majors. We use Replit, as far as...so it's easy for me to just run their code as is. I create a project in it; they just run it; they can't have an excuse. 'I couldn't download this.' 'I don't have space.' 'I don't have a computer.' It's all browser-based. So it takes out that 'I can't test your code because you're cheating.'*

They also spoke about the importance of testing and how they made that part of the course. They mentioned it could be something they encouraged the students to do or that they were involved in as well. As one educator detailed:

> *Everything has some sort of solution that I have to check. And sometimes I ask them to check a peer to see if it makes sense. That has a lesser degree of...it works, then. . . The efficiency of that is, that it's quick because it doesn't take too much time on my end, but they don't do a good job of grading each other. So, every exam has some sort of coding requirement in which we have to test it. And ideally, I do it after the fact, like after an exam after a midterm. I do it with them, and I show them that it can be done in the timeframe that I give them. Everybody is very skeptical about that. But that's the extent, at least [for] my institution.*

Problem decomposition was something that often required multiple efforts and ways to approach it in lessons. The participants talked about how it entailed more than just teaching the foundations, but considering ways to break things down step by step. As someone highlighted:

> *What I tend to do is, especially in my intro-level classes, is like, I'll give like a proper coding prompt, and then copy and paste it incorrectly and make them as comments. So, I try to get them to use comments a lot. And so we'll go line by line, "What do we need to do to just accomplish this one thing, instead of trying to attack it all at once?" And I'll, of course, help my students, when we do, like, in-class work as an exercise. And they'll be like, 'it's wrong.' And I'll be like, 'Well, this is what the comment is,' I mean, 'this is what the error says; what do you think this means?' And I'll be like, 'What is that?' And 'What is that?' And then they're like, 'Oh, it's not the right datatype.' And then I'm like, 'What should it be?' And then they'll figure it out.*
>
> *So, a lot of times, especially with the beginner students, they can't, because they're new, they can't figure out the error codes or whatever that Replit gives, even though Replit does a really good job of pointing them out, but they just still can't understand it. So me, breaking down what the code means, and they'll have that 'Aha!' moment, and they can figure it out. Usually it's spelling though, and so they just have to keep looking at it again and again.*

Others mentioned trying to ask students to talk through errors or to write comments to themselves. One faculty member spoke about how they deliberately use comments as a way of tracking the thought process. They stated:

> *What I do, and I'm trying to push it on the faculty, particularly [...] Do not erase your code. Comment it out. I want to know what you were thinking. And I'm also trying to let them understand that, if that was what they first thought, maybe when they see their comments a month from now, they will understand that that wasn't the right way, and this is the process they went through. Eventually, it gets easier for them. [...] Everybody erases because they don't want to show weakness or their errors to anybody, which is a pity.*

Faculty also referred to things that fell under the theme of "Job Preparation," incorporating training or lessons geared towards the hiring process in computing. They mentioned potentially going through pretend scenarios to help establish a more realistic encounter. As one person described:

> *...actually have them practice mock interviewing, because a lot my students like, we'll do LeetCode all day, but it's not the same when you get actually in front of somebody interviewing you. So actually, practicing the entire thing and not just problems, problems, problems. Because I think one thing, also, is that with my students, especially students in general, anxiety as far as testing anxiety or even interview anxiety, and so not experiencing that until you're in the moment, that's no good.*

Meanwhile, several other educators noted that they would work with career services on their campus to aid students with preparing application materials like resumes or portfolios. Another participant mentioned a more unique approach that combined the creation of such content with more specific roles. The instructor noted:

> *I would take students on field trips to companies like [Company Name], all these engineers (it was STEM-based), and I would read, have them to, not just tour the place, but have the VPs and the upper-level admins speak to the students about their process and how they got to where they were and allowed them to ask questions. And then when we would leave, I would create mock interviews every semester. I didn't mandate it, but I would encourage the students to do it, and I would create a mock interview around whatever company we just left. I would go to their site, find an entry-level position, and the students would have to create cover letters, resumes.*

### *5.2 RQ2: What actions do educators see as necessary for institutions to take to enhance students' technical interview performance and career attainment?*

As shown in Figure 2, we observed two over-arching themes in the data: "Supportive Practices" and "Technical Interview and Career Knowledge." These themes were compromised of seven codes. Again, we elaborate further on each of these below.

Table 2: RQ2: Resulting themes and codes

| **Theme** | **Code** | **Description** |
|---|---|---|
| *Supportive Practices* | Encouragement | Offering support and reassurance |
| | Sharing Information | Providing information or suggestions, or raising awareness of technical interviews |
| *Technical Interview and Career Knowledge* | Student Courses | Course focusing on problem solving, applying critical, and/ or creative thinking |
| | Campus Organizations | Groups, clubs, or teams on campus that provide preparation or services around career development |
| | Industry Partnerships | Discussions around industry representatives, organizations, and/ or companies working with institutions to provide students with details on roles, preparation, or that helped them to network |
| | Instructor Training | References to instructors being unfamiliar with the process themselves, not having been through technical interviews, or the benefits to offering familiarity and practice for them |
| | External Services | Mentions of using established platforms, organizations, and/ or groups that offer preparation and mock interviews |

The theme of "Supportive Practices" referred to the many ways educators envisioned students being prepared for the hiring process and encouraging their persistence in the field. Many participants referred to emotional support and encouragement they would give or observe their colleagues giving, such as "*So tell them, 'Don't be afraid.*'" Another educator described that:

> *I encourage them, like even if they don't think they're prepared, at least go and try because at least you'll have a feel for what other ones will be like and so, hopefully, you can use this as a learning experience.*

Others elected to provide information and awareness of the process. As one educator detailed:

> *I know, one of the complaints my students have about the technical interviews is that if we are coding in real life, we can just look stuff up, and so they don't like the being*

> *put on the spot-ness of it. I mean, I don't know how to incorporate that because I was trying to tell them, we're talking about like, ChatGPT and how AI is taking jobs and things like that. And I was like, 'People while you can still put things out, people still have to know that you have that thinking process, that critical thinking process. So that's why these technical interviews are important and necessary.' And some of them were like, 'Oh okay, I get it, I see it.' But yeah, they're just like, 'We wouldn't have to do this in the real world. If I need to look up the problem I can get on Stack Overflow.' But I don't know how you alleviate that in the initial interview process. Because I do think the technical interview is useful, if that's some of the criteria.*

In addition, the theme of "Technical Interview and Career Knowledge" referred to the variable approaches taken at institutions to help students gain familiarity with hiring practices, potential jobs, and what to expect from technical interviews. The educators talked about options to further hone students' skills and provide more practical training. While courses around technical interviews specifically were not available, they did have more information around general employability. For example, one participant shared:

> *So [for] all three [computing degree] concentrations, we have a course like "Readiness course." [...] the readiness course will cover all the technical skill, what they have learned so far, and they will get a chance to work on a real-time project as well. So, as I mentioned, the company, Open Avenues, so, what they do, they will provide us the project, a real time company project, for this course. So, the student can work on this course, then the student can learn for, let's say like maybe a small software development modules, they can work on for the scope [of the] course. So, they will learn about what are the main program languages they have used or what framework they have used, the used framework or Java or they are using any specific under like the MVC [Model, View, and Controller] framework to develop the coding and everything. So, they will get a chance to learn all the technical aspects. And, you know, I told you last time, like, we are providing career seminar, the career seminar course will offer all the soft skills like how they can prepare for the interview, how they can face the interviewer and everything. So, that this kind of course, the readiness course, we will cover more of the technical aspects, the seminar course will cover the interview preparation aspects of the questions and everything.*

However, often, instructor training was lamented as being limited, since they went directly into their academic positions. Frequently, the educators cited this as a problem which could prevent faculty members within departments from incorporating content into lessons more broadly. It was emphasized that:

> *It's not actually done inside the class, and one reason is because it's the professor, right? If the professor knows it, if they've been in industry, yes, they can do it. But if they haven't, you're asking them to do a skill they don't know themselves.*

Despite such concerns, many participants pointed to groups, teams, and clubs on campus that would host events or provide training for students. As one person commented, "*And it will probably for, on our campus, have to come through one of our student organizations, and we do have many.*" Several others mentioned career services, or a writing-focused aid, like "*we do have* *those type of, we have groups on the campus that, like our writing center, I guess I should say, that would help them with those types of things, resume writing, cover letters.*"

Industry partnerships with companies and organizations were seen as other ways of providing training and opportunities. While only a select few had such options, others mentioned they wished they did. One participant shared that:

> *We are partnering with a nonprofit organization called Open Avenues. So what they do, they will have information sessions for attending interviews, and making connections with industry people if it's something like that. So that is also helpful for our students to learn more about what's going on in the industrial side.*

Meanwhile, external services and opportunities were cited as being preferable, although educators mentioned that such options may be limited at their own institution. Those that did have such programs or alliances articulated how valuable they could be for students' preparation. As one participant described:

> *...all our students have access to Big Interview [a step-by-step job interview training software]. However, a lot of professors push it so it's actually an assignment in every single class I have. I am gonna see the video...Big Interview has a bunch of technical questions where you record yourself and then you can see how you record and you can also re-record yourself so you continuously practice right, you know. So, for example, one of the questions may be... there's... it's broken up into different sections. There's general, just your general interview questions. Then, there's soft skills, and there's technical customer service. And then, on the programming side, it's technical questions on programming. You know, more not to actually do it, but to say, 'What is an array?' 'What is a string?' 'What, you know?' It's, these kinds of questions are 'How would you explain that?' You know, one of the interviews actually, 'How would you explain that to my nine year old child?' You know, so those are the kinds of questions they have, then they record it. And the interview is not being written, it's being spoken to them. So it's a back and forth. So you have to have a camera and you can record it. And if they can't record it, we just have them go into our computer lab and they make cameras available and record it there. But I always have it as a, 'You're gonna get 10 points just by doing this. And if you miss this, you're losing 10 points.' It's just always there.*

## 6 Discussion

Findings from this exploration are multi-dimensional. In this section, we elaborate further on the results as they pertain to each of our RQs.

### *6.1 RQ1: How do educators incorporate learning experiences for undergraduate computing students into courses to aid in their career outcome expectations?*

The participants in our focus group mentioned that obtaining a computing position was often the students' priority, as described by "Choice Goals" in SCCT. In order to help them achieve this and aid in career outcomes, the faculty employed a variety of approaches in their classrooms. Beyond merely trying to impart the required knowledge, the educators included activities and assignments that made skill development and cultivating professional dispositions a more deliberate goal.

As emphasized by the theme “Fostering Professional Skills,” promoting interpersonal dynamics and other non-technical capabilities was seen as critical for students’ long-term success. Such findings align with the work of others, who emphasized their importance and also noted that professional skills may be rated even higher than technical skills in terms of expectations from a new team member [32]. However, the educators cautioned that students may lack confidence in their social interactions, so providing them with ways to engage in higher education could help to build that confidence, fitting with the connection between learning experiences and self-efficacy in SCCT. Although, as Bandura has distinguished, confidence may refer to a strength of conviction for a belief but may not include its direction, whereas self-efficacy does suggest a focused goal [28, 33].

Providing students chances to work on verbalizing their strategy for finding solutions, either alone or with peers, was often articulated by the participants as helpful to enhancing communication. The faculty also spoke about how having students solve problems together and rotating who they partnered with could be valuable to gaining experience with teamwork. They highlighted that switching teams could not only give the students exposure to completing tasks with different populations and their unique approaches, but that it could also serve to provide support for them as minoritized students saw others like themselves. These observations dovetail with research that suggests reducing isolation can be beneficial to persistence in computing [34].

Given that computing does have inherently technical components as well, educators mentioned how they may try varying techniques to engage students and help them tackle problems in the domain, as described by the theme of “Fostering Technical Skills.” Command-driven labs in virtual environments and tools like Replit would help to eliminate concerns students had around space. Software testing and individual testing for code were integrated by several instructors in lessons and made part of the exams. The instructors also spoke to how working through the solutions themselves, both in class and when reviewing the exam solutions, was important to show students they could be completed in a timely way. Such an approach corresponds with what others have described as valuable for teaching students how to plan algorithms and code through watching videos of experts offering “live-coding-based instruction” [35].

Another component of technical comprehension seen as critical was incorporating verbalization of the coding thought process into practice in the context of troubleshooting. Instructors mentioned how getting things wrong could be a learning experience, even more so than getting the correct answer. Asking students to walk them through the choices they would make, code tracing, as they solved problems together line-by-line, could help them recognize errors in an approach or catch indentation problems. Such suggestions are supported by research on how comprehension, activities based on evidence, and going through workflows can correct flawed mental models to enhance debugging for novice programmers [36].

Turn-taking and working together to tackle problems could also aid in working through solutions in the way expected during hiring, as described by the theme of “Job Preparation.” Mock interviews, or having one play the interviewer and the other the candidate and eventually switching roles, were seen as beneficial to break problems down and find an answer. Such practices were seen as especially helpful when paired with experts (e.g., instructors or teaching assistants) walking around to provide guidance.

### 6.2 *RQ2: What actions do educators see as necessary for institutions to take to enhance students' technical interview performance and career attainment?*

Beyond what was offered within courses, career support and preparation were described as coming through many different avenues at institutions. The educators envisioned this as a combination of actions taken more broadly and the efforts of individuals, groups, and outside partnerships. For example, Open Avenues Foundation[2] provided everything from corporate projects for students to connections with industry professionals.

Such opportunities could be especially beneficial in cases where students might doubt their own abilities or where they may not see the connections between what their curriculum covers and what they might need to demonstrate later in technical interviews and the workplace. The theme of "Supportive Practices" referred to finding ways to bolster students' self-efficacy as it may impact their longer-term goals (from SCCT) in the field. Educators mentioned how frequently students would report feeling removed from certain classes, like data structures, after completing them. As such, the students struggled and were frustrated when asked about material from them later and expected to transfer the knowledge.

Instead of assuming students could or would recognize such information would be needed later, when instructors relayed the ongoing value and importance of applying these to their praxis to improve their programs, it was seen as something that could help students recognize them as more than just ideas related to a specific class. Heaps and stacks could be seen as part of their understanding and application for more advanced programs that they could use in their future jobs. While such linkages may seem obvious to faculty, verbalizing this more explicitly is something students may benefit from.

The educators also articulated how, in computing, it is important to be self-directed and willing to seek out new solutions, technologies, and programming languages. This aligns with findings from interviews with computing professionals, who mention that one of the most critical dispositions is a "lifelong learning orientation" [16]. This is considered an ability "to learn new knowledge and/or skills to further their career in computing or for work needs" [16, p. 278]. Such a curiosity to acquire new knowledge and skills may not be something that is taught in coursework, although the authors have noted it may be possible. However, the value of continuously working to understand new systems and programming languages is something that could be encouraged in classrooms.

Students also developed "Technical Interview and Career Knowledge" through different sources, even without formal courses geared towards such preparation. Educators mentioned how workshops, guests, and seminars on campus could all be used to develop interests around varying roles and aid in the choice actions around preparation. Whether through external industry speakers or in the form of helping with resumes, such information and training were seen as important for students to ultimately obtain a job, as described by performance domains and attainments in SCCT.

[2] https://www.openavenuesfoundation.org/

Also, they suggested that using existing services (e.g., Big Interview[3], Karat's Brilliant Black Minds[4]) could also help students feel more prepared for the hiring process and technical interviews in particular. Such services could provide a more formal approach rather than going through the process with peers, which could be more realistic in terms of what students may encounter. The instructors mentioned that such partnerships and raising awareness of such programs could be another way for students to feel better equipped for choice actions.

Something that should be considered is that many educators did report being unfamiliar with technical interviews and what they entailed, or knowing that others at their institution were. Given that this may limit how much educators can help students prepare, it may be worth providing training to faculty. Whether through industry partnerships to offer workshops or to directly administer mock interviews, providing such experiences for educators could ultimately help to foster empathy for what the hiring process may entail and could raise awareness of the expectations for those who may be unfamiliar with it. Education can be valuable and can empower and equip faculty to better aid their students as well.

## 7 Limitations and Future Directions

The study conducted a focus group of a subset of individuals across the United States. However, the experiences and voices represented may not be reflective of all institution types or of all educators. There is a breadth of diversity both in institutional types, geography, structures, expectations, programs, and faculty roles, among other features. Varying levels of familiarity with technical interviews may further influence what is said. Going forward, we suggest expanding not only the quantity of educators involved, but also the voices of those present, as it could yield additional insight.

Another consideration is that we did conduct a focus group, which may influence how individuals respond or lead to "group think" in respondents. Prior literature has described that a key facet of quality is reaching saturation [37]. Since we did not conduct multiple focus groups, we cannot confirm that we reached meaning saturation. This is something that could be considered through the expansion of the focus groups or through semi-structured interviews at additional institutions. However, we did reach code saturation based on code identification and the stability of the codebooks to identify the variability within the data collected.

## 8 Conclusion

In this paper, we described a pilot conducted to better understand educators perspectives on the hiring process in computing and potential ways to integrate preparation into existing institutional structures. We observed that many educators may be unfamiliar with technical interviews, not having completed any themselves. As such, it could be valuable to provide training for faculty to better prepare their students. Moreover, as spoken to by the theme of technical interview and career knowledge, given that training frequently occurs through different organizations on campus, there is an opportunity to further extend such lessons into coursework so that students can gain familiarity at multiple access points. Moreover, as spoken to by supportive practices, it is helpful to consider how institutions can further provide information and encouragement as students set goals and begin to take their next professional steps. Although higher education is not

[3] https://www.biginterview.com/

[4] https://karat.com/brilliant-black-minds/

intended to serve as a trade school, through collaborative efforts with industry and academia, there are a lot of routes through which we can empower students and help them achieve career attainment and long-term success.

## Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF), Award # 2137394. Any findings, conclusions, and recommendations expressed in this work do not necessarily reflect the views of the NSF. Also, we would like to thank the STARS Computing Corps.

## References

[1] U.S. Bureau of Labor Statistics, "Computer and Information Technology Occupations: Occupational Outlook Handbook," April 2024. [Online]. Available: https://www.bls.gov/ooh/computer-and-information-technology/home.htm

[2] G. M. Lundberg, B. R. Krogstie, and J. Krogstie, "Becoming Fully Operational: Employability and the Need for Training of Computer Science Graduates," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020, pp. 644–651.

[3] G. L. McDowell, *Cracking the Coding Interview: 189 Programming Questions and Solutions*, 6th ed. CareerCup, 2015.

[4] L. Aline, "We analyzed 100k technical interviews to see where the best performers work. here are the results." March 2022. [Online]. Available: https://interviewing.io/blog/we-analyzed-100k-technical-interviews-to-see-where-the-best-performers-work-here-are-the-results

[5] 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)*. IEEE, 2019, pp. 1–9.

[6] Z. Nagy, *Soft Skills to Advance Your Developer Career*. Springer, 2019.

[7] 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.

[8] S. Lunn, M. Ross, Z. Hazari, M. A. Weiss, T. Solis, M. Georgiopoulos, and K. Christensen, "Uneven Playing Field: Examining Preparation for Technical Interviews in Computing and the Role of Cultural Experiences," in *Proceedings of the American Society for Engineering Education (ASEE) Annual Conference & Exposition*, 2021.

[9] S. Sentance and A. Csizmadia, "Computing in the curriculum: Challenges and strategies from a teacher's perspective," *Education and information technologies*, vol. 22, pp. 469–495, 2017.

[10] Y. Qian and J. Lehman, "Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review," *ACM Transactions on Computing Education*, vol. 18, no. 1, pp. 1–24, October 2017. [Online]. Available: https://doi.org/10.1145/3077618

[11] L. K. Baartman and E. De Bruijn, "Integrating knowledge, skills and attitudes: Conceptualising learning processes towards vocational competence," *Educational Research Review*, vol. 6, no. 2, pp. 125–134, 2011.

[12] F. F. Patacsil and C. L. S. Tablatin, “Exploring the importance of soft and hard skills as perceived by it internship students and industry: A gap analysis,” *Journal of Technology and Science education*, vol. 7, no. 3, pp. 347–368, 2017.

[13] E. Holloway and J. S. Linvill, “Changing the conversation surrounding students’ professional skills: Making the case for the importance of professional skills, and more inclusive language,” in *Proceedings of the 2023 ASEE Annual Conference & Exposition*, Baltimore, MD, USA, June 2023.

[14] W. Groeneveld, B. A. Becker, and J. Vennekens, “Soft Skills: What do Computing Program Syllabi Reveal About Non-Technical Expectations of Undergraduate Students?” in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE ’20)*. Trondheim, Norway: ACM, June 2020, pp. 287–293.

[15] S. Frezza, T. Clear, and A. Clear, “Unpacking Dispositions in the CC2020 Computing Curriculum Overview Report,” in *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2020, pp. 1–8.

[16] D. Tagare, S. Janakiraman, M. Exter, S. Duan, M. Sabin, and J. Tavakoli, “Dispositions that Computing Professionals Value in the Workplace: Systematic Literature Review and Interviews with Professionals,” in *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*. Chicago, USA: ACM, August 2023, pp. 270–283.

[17] S. Duan, M. Exter, D. Tagare, and M. Sabin, “Managerial and Professional Skills and Dispositions from Professionals’ Interviews,” in *Proceedings of the 23rd Annual Conference on Information Technology Education*, Chicago, IL, USA, September 2022, pp. 1–8.

[18] N. Kiesler and J. Impagliazzo, “Industry’s Expectations of Graduate Dispositions,” in *Proceedings of the 2023 IEEE Frontiers in Education Conference (FIE)*. College Station, TX, USA: IEEE, October 2023, pp. 1–5.

[19] M. Behroozi, S. Shirolkar, T. Barik, and C. Parnin, “Does Stress Impact Technical Tnterview Performance?” in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Virtual Event, USA, November 2020, pp. 481–492.

[20] M. Behroozi, S. Shirolkar, T. Barik, and C. Parnin, “Debugging Hiring: What Went Right and What Went Wrong in the Technical Interview Process,” in *Proceedings of the 2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. Seoul, Republic of Korea: ACM, December 2020, pp. 71–80.

[21] S. Lunn, M. Ross, Z. Hazari, M. A. Weiss, M. Georgiopoulos, and K. Christensen, “The Impact of Technical Interviews, and other Professional and Cultural Experiences on Students’ Computing Identity,” in *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2021)*. Virtual Event, Germany: ACM, New York, NY, USA, 2021, pp. 415–421.

[22] S. J. Lunn, E. Zerbe, and M. Ross, “Need for Change: How Interview Preparation and the Hiring Process in Computing Can Be Made More Equitable,” in *Proceedings of the 2022 CoNECD (Collaborative Network for Engineering & Computing Diversity)*, New Orleans, LA, USA, February 2022.

[23] S. J. Lunn, E. Zerbe, and M. Ross, “You’re Hired! A Phenomenographic Study of Undergraduate Students’ Pathways to Job Attainment in Computing,” *ACM Transactions on Computing Education*, vol. 24, no. 1, pp. 1–29, 2024. [Online]. Available: https://doi.org/10.1145/3636514

[24] A. Kapoor and C. Gardner-McCune, “Introducing a Technical Interview Preparation Activity in a Data Structures and Algorithms Course,” in *Proceedings of 26th Annual ACM Conference on Innovation and Technology in Computer Science Education conference (ITiCSE’21)*. Virtual Event, Germany: ACM, New York, NY, USA, 2021, pp. 633–634.

[25] E. Dillon, A. Dina, M. McMichael, T. Wimberly Jr, L. Brown, and K. L. Williams, “Exposing Early CS Majors to Technical Interview Practices in the Form of Group-Based Whiteboard Problem Solving Activities,” in *Proceedings of the 2023 ASEE Annual Conference & Exposition*, Baltimore, MD, USA, June 2023.

[26] R. W. Lent, S. D. Brown, and G. Hackett, “Social Cognitive Career Theory,” in *Career Choice and Development*, 4th ed. Jossey-Bass, 2002, pp. 255–311.

[27] R. W. Lent, A. M. Lopez Jr, F. G. Lopez, and H.-B. Sheu, “Social cognitive career theory and the prediction of interests and choice goals in the computing disciplines,” *Journal of Vocational Behavior*, vol. 73, no. 1, pp. 52–62, 2008.

[28] A. Bandura, “Social foundations of thought and action,” *Englewood Cliffs, NJ*, vol. 1986, no. 23-28, 1986.

[29] V. Braun and V. Clarke, *Thematic analysis*. American Psychological Association, 2012.

[30] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[31] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, pp. 159–174, 1977.

[32] C. Cortázar, I. Goñi, A. Ortiz, and M. Nussbaum, “Are Professional Skills Learnable? Beliefs and Expectations Among Computing Graduates,” *ACM Transactions on Computing Education*, vol. 24, no. 2, pp. 1–25, 2024. [Online]. Available: https://doi.org/10.1145/3641551

[33] A. Bandura, “Perceived self-efficacy in the exercise of personal agency,” *Journal of applied sport psychology*, vol. 2, no. 2, pp. 128–163, 1990.

[34] W. Aspray and W. Aspray, “Building Educational Infrastructures for Broadening Participation in Computing,” *Women and Underrepresented Minorities in Computing: A Historical and Social Study*, pp. 223–259, 2016.

[35] Y.-T. Lin, M. K.-C. Yeh, and S.-R. Tan, “Teaching Programming by Revealing Thinking Process: Watching Experts’ Live Coding Videos With Reflection Annotations,” *IEEE Transactions on Education*, vol. 65, no. 4, pp. 617–627, 2022.

[36] J. Whalley, A. Settle, and A. Luxton-Reilly, “A Think-Aloud Study of Novice Debugging,” *ACM Transactions on Computing Education*, vol. 23, no. 2, pp. 1–38, June 2023. [Online]. Available: https://doi.org/10.1145/3589004

[37] M. M. Hennink, B. N. Kaiser, and M. B. Weber, “What Influences Saturation? Estimating Sample Sizes in Focus Group Research,” *Qualitative Health Research*, vol. 29, no. 10, pp. 1483–1496, 2019.

## Appendix

We present the script employed during the focus group in Table 3.

Table 3: Script for the focus group

| |
|---|
| Welcome. My name is______.<br>We want to thank you for your participation in this workshop. Before we get started, we would like to begin by getting to know each other, and having a discussion around technical interviews, including our experiences, and potential suggestions on how we can better prepare students for the next stage in their career. If you are willing, we would like to record the interaction so that we can use the feedback from what we will refer to as a "focus group" to consider new opportunities for students and faculty development. Do I have your consent to record the interview?<br><<Wait for verbal consent or obtain written consent.>><br>During this focus group, I will ask you questions about your experiences as a computing educator. The focus group will take between 45 and 60 minutes. There are no right or wrong answers, and the purpose of this focus group is to learn from you as you are the expert of your experiences. You don't have to answer any questions if you don't want to, but I will encourage each person to share. Feel free to ask me to repeat or clarify any questions.<br>Do you have any questions for me before we begin? |
| What have you heard from students at your institution about the hiring process in computing and/or their experiences? |
| What skills or competencies do you think are important to be successful in a technical interview? |
| How much of a role do you feel that educators should play in preparing students for technical interviews? |
| What aspects of the technical interview process or preparation for the technical interview process does your:<br>- Department offer students?<br>- College or school offer students?<br>- Institutions offer students?<br>- Organizations on campus offer students? |
| What learning experiences may be part of your curriculum that you think could be most beneficial to understanding technical concepts, and cultivating technical skills or competencies? |
| What learning experiences may be part of your curriculum that you think could be most beneficial to cultivating non-technical skills or competencies? |
| What do you think could be most beneficial to helping students prepare for technical interviews?<br>- Where do you think this may best fit in your curriculum? |
| What do you think could be most beneficial to helping students with problem decomposition?<br>- Where do you think this may best fit in your curriculum? |
| What do you think could be most beneficial to helping students test their solutions to programming problems?<br>- Where do you think this may best fit in your curriculum? |
| What do you think could be most beneficial to helping students verbally express themselves while problem solving?<br>- Where do you think this may best fit in your curriculum? |
| How do you think we can better help students to achieve their professional goals? |
| What do you think could be a better way for companies or organizations to evaluate candidates for computing positions? |
| Do you have any other feedback on technical interviews or the role of education in workforce preparation? |