1. **Improving Computer Science Curricula to Better Prepare Students for Technical Interviews**

Current computer science curricula often lack sufficient focus on the specific skills and techniques required for technical job and internship interviews, particularly in areas such as algorithms, data structures, time complexity, and whiteboard problem-solving (Dillon et al., 2023). This gap leaves students underprepared for the rigorous and specialized nature of these interviews, affecting their confidence and performance in securing job opportunities.

**2. Criteria**

The below criteria showcase some requirements that can be used to evaluate any potential solution to the issue of students being unprepared for technical interviews throughout their journey of the Computer Science curriculum.

| **Required Criteria** | **Preferred Criteria** |
| --- | --- |
| Core knowledge of technical interviews such as time complexity, data structures, algorithms, and problem-solving should be covered | Should be integrated with the industry through input and collaboration from professionals |
| Should be feasible and practical to implement in the curriculum | Cater to different learning styles and schedules |
| All students should have access | Easily scalable to accommodate increasing class sizes |
| Must effectively improve preparedness for technical interviews | Improve pass rates of technical interviews |

**3. Options**

1. Dedicated Interview Preparation Course: Design a course that specifically teaches and prepares students for technical interviews. This course would cover common topics such as algorithms, data structures, time and space complexity, and problem solving techniques, but also expand more than a standard Data Structures course (Field et al., 2023). These extra topics should include algorithmic thinking under pressure and a time limit, speaking confidently while coding, and whiteboard ability for solutions, which are practical skills required for any technical interview (Field et al., 2023). Specific assignments should be Leetcode and HackerRank timed coding questions, for which they would self-record their thought process while solving and afterwards, analyze their own performance and how they can improve (Dillon et al., 2023). This self-awareness is critical for targeted skill development and overall preparedness (Kapoor et al., 2023). A key component of this course should be mock technical interviews (Kapoor et al., 2023). These should be frequent assignments so that students are habituated to them and get a boost in confidence. Initially, these interviews should be arranged between peers with specifications given for minimum time, question difficulty level, level of assistance allowed, etc. Ultimately, the interviews should be conducted by the Teaching Assistants of the course, to simulate a more professional and higher-stakes atmosphere. The structured practice and feedback provided by these activities will be effective in reducing the uncertainty and fear often associated with technical interviews, making students feel more prepared and self-assured (Kapoor et al., 2023).
2. Peer-Led Study Groups and Mentorship Programs: It is imperative that students are more eager to participate in an arrangement that they can personally relate to, and aim for a station that they can see themselves at (Deibel, 2005). Students who have just gone through the process and passed are the perfect pool of mentors. Peer support and mentorship can create a collaborative learning environment, offering personalized guidance and encouragement (Deibel, 2005). For any student, knowing they have a support system helps reduce anxiety about their preparedness for technical interviews and teamwork in professional settings. Student organizations should be asked to expand their mentorship programs from purely social to career-oriented and technical and establish peer-led study groups and mentorship programs where senior students and alumni can guide juniors in their interview preparation (Craig Michelle et al., 2018). These groups would provide regular practice sessions, problem-solving workshops, and mentorship. This would benefit both groups, as for the mentors it would enhance communication skills and confidence, and for the mentees it would increase their practical knowledge (Deibel, 2005). In addition, both groups would gain connections and increase their professional networks, which is crucial for any career (Craig Michelle et al., 2018). During the implementation of this solution, sufficient resources should be allocated by the institute to these student organizations. As a bonus to just interview preparation, interacting with peers and mentors helps students practice identifying the right person to ask for help and framing their questions effectively (Deibel, 2005). This mirrors the real-world scenario where developers often need to consult with different team members to navigate complex codebases.

**4. Solution**

Dedicated Interview Preparation Course: Implementing a dedicated interview preparation course for computer science students involves designing the course content with industry collaboration, training faculty, and integrating the course into the curriculum. This approach ensures students gain essential technical interview skills and are well-prepared for the job market (Craig Michelle et al., 2018). The solution meets the required criteria by covering core knowledge areas, being practical to implement, accessible to all students, and improving preparedness for technical interviews. It also meets preferred criteria by integrating industry input, catering to different learning styles and schedules, being scalable, and aiming to improve technical interview pass rates.

*Step One: Designing the Content*

This step will take at least six months. Partner with industry professionals to co-design a curriculum that aligns with the latest industry standards, ensuring students gain relevant skills and knowledge. Focus on essential topics like algorithms and data structures, and include practical sessions such as coding challenges and mock interviews for hands-on experience. Implement assessment tools that mimic real technical interviews, like timed coding tests and peer reviews, to track progress and provide targeted feedback.

*Step Two: Faculty Training and Resource Allocation*

This step will take at least three months to complete. Provide training for faculty on the new course content and teaching methods, ensuring they stay updated with industry practices. Allocate resources like equipped classrooms, online coding platforms, and enough staff to manage the course. Establish a feedback loop to continually improve the course based on input from students and industry partners, keeping it relevant and effective.

*Step 3: Integration into the Curriculum*

Get the new course approved as an elective or mandatory component and schedule it to avoid conflicts, ensuring it's an integral part of the curriculum and flexible for students. Promote the course through information sessions and collaboration with student organizations, highlighting career benefits. Monitor student performance and confidence using surveys and performance data to evaluate and improve the course, ensuring it remains effective and relevant.

References

Kapoor, A., Panchal, S., & Gardner-McCune, C. (2023). Implementation and Evaluation of Technical Interview Preparation Activities in a Data Structures and Algorithms Course. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, *1*, 882–888. <https://doi.org/10.1145/3545945.3569755>

Deibel, K. (2005). Team formation methods for increasing interaction during in-class group work. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education - ITiCSE ’05*, *37*(3), 291–295. <https://doi.org/10.1145/1067445.1067525>

Craig Michelle, Conrad Phill, Lynch Dylan, Lee Natasha, & Anthony Laura. (2018). Listening to early career software developers. *Journal of Computing Sciences in Colleges*, *33*(4), 138–149. <https://doi.org/10.5555/3199572.3199591>

Field, R., Fuller, S., & Dillon, E. (2023). Surveying the Importance of Integrating Technical Interviews into Computer Science Curriculums and Increasing Awareness in the Academy. *2023 ASEE Annual Conference & Exposition*. <https://doi.org/10.18260/1-2--44382>

Dillon, E., Dina, A., McMichael, M., Theodore Wimberly Jr, Brown, L., & Williams, K. (2023). Exposing Early CS Majors to Technical Interview Practices in the Form of Group-Based Whiteboard Problem Solving Activities. *2023 ASEE Annual Conference & Exposition*. https://doi.org/10.18260/1-2--43665