Got it! A one-page write-up can really help explain your intentions, showcase the thought process behind your innovative approach, and invite fellow instructors to reflect on their own teaching practices. Here’s a draft structure for a concise yet impactful position piece:

“Committing to Connection: A New Approach to Engagement in Technical Learning”

Introduction

In many technical courses, students often feel isolated, unsure of their abilities, and disconnected from their peers. This can lead to low motivation and a belief that they’re “not cut out” for tech. To counter this, I designed an introductory GitHub exercise that not only teaches technical skills but also builds community, accountability, and confidence. This approach aims to meet students where they are and help them see themselves as capable contributors from day one.

Why This Approach?

1. Changing Beliefs Through Social Learning:

Research shows that when students share goals, struggles, and achievements, they are more likely to internalize positive beliefs about their own abilities. My activity uses peer connections to anchor GitHub practices in mutual support rather than individual pressure.

2. Motivation Through Relevance and Small Wins:

Traditional tech exercises often focus solely on syntax or workflows, which can feel abstract. By tying commits to personal introductions and goals, students associate progress with something meaningful. This also provides early, achievable wins that make continued learning more rewarding.

3. Social Accountability and Confidence:

When students publicly “commit” to a goal and log their collaborations, they experience both a technical and emotional investment in the journey. Instead of fearing GitHub as an intimidating tool, they see it as a space to document their growth and contributions—just like professional developers.

Why It Matters for Instructors Too

As instructors, we’re not just teaching technical steps—we’re facilitating a shift in mindset. This lesson isn’t just about GitHub; it’s a way to model how learning is an evolving, communal experience. Implementing this approach ourselves reminds us to:

• Commit to our own growth and try new methods.

• Connect with students and colleagues in meaningful ways.

• Celebrate progress rather than perfection.

Conclusion

This activity reframes a technical skill as a story of connection and growth. By fostering collaboration and social accountability, we not only teach programming but help students believe in their ability to thrive. As instructors, adopting this lens in our own practice can strengthen our teaching and help our students feel empowered—not just as learners but as future contributors.

This keeps it concise while touching on the why, how, and impact of your approach. Does this tone and structure feel like it aligns with your vision?

Yes. You’re definitely using that strategy effectively! Your “Git Commit Your Journey” lesson plan incorporates several motivational principles aimed at fostering collaboration, peer accountability, and intrinsic motivation. Here’s a breakdown of why your approach aligns well with the strategies for changing beliefs and fostering motivation:

What You’re Doing Well:

1. Belonging and Peer Support

• By encouraging students to introduce themselves and connect with others, you’re lowering the social barrier and creating a sense of community—a proven way to boost engagement and reduce the fear of failure.

2. Anchoring Motivation Through Social Accountability

• The “commit” metaphor ties personal goals to tangible actions within GitHub. By making their learning journey public and collaborative, students are more likely to stay engaged.

3. Autonomy and Choice

• Asking students to “commit” to a personal learning goal gives them ownership. They’re choosing what matters to them, which strengthens intrinsic motivation.

4. Positive Reinforcement

• Encouraging responses with positive language (“I like how…”) builds a supportive atmosphere and models constructive feedback. This can help shift students’ beliefs about feedback from judgmental to helpful.

5. Small Wins and Clear Actions

• Activities like adding peers to milestones.txt or README.md provide quick, tangible steps that build competence, reinforcing a “progress over perfection” mindset.

Suggestions for an Even Greater Impact:

1. Reflection Prompts

Add a reflection step where students write a short note on how the collaboration changed their view of GitHub or teamwork. For example:

• “What surprised you about working with your peers today?”

2. Reframing Failure

Encourage students to write their first commit even if they’ve made mistakes. Reinforce the idea that a messy commit is still progress—similar to how developers iterate.

3. Peer Celebrations

Consider adding a “shout-out” board or an optional place to post about milestones they’ve achieved (even small ones). Publicly celebrating can reinforce that the class is about progress, not perfection.

Your lesson already fosters connection and motivation effectively, and these small additions could help reinforce students’ belief that learning is a shared, positive experience. What do you think—are there any areas you’d like to focus on improving further?