

Topic 1.1: Collaboration

LO: CRD-1.A, CRD-1.B, CRD-1.C | Skill: 1.C, 6.A | Scenario: 1A: The TideTrack App Team

Teacher Key intro

Use the model responses below to guide the Day 2 scenario debrief. Don't simply read them aloud — prompt students to defend their own answers first, then validate or extend with the model. Look especially for students who can explain WHY diverse perspectives matter, not just state that they do.

Scenario — The TideTrack App Team *(Teacher Key)*

Original scenario for AP Computer Science Principles Topic 1.1. Not based on any real team or product.

Background

Four students at Bayside High are building TideTrack, a free app that shows local tide times and safe walking windows for a popular coastal trail. The team is Priya (strong at design), Marcus (the most experienced coder), Lin (who grew up walking the trail and knows it well), and Devon (who is new to coding but great at organizing the group).

The situation

Two weeks before the school showcase, the team meets to make decisions. Marcus has written most of the code so far, mostly on his own and late at night, and the others are not sure how it works. Priya has mocked up a clean design but Marcus thinks it will be slow to build. Lin keeps pointing out that the trail floods in spots the app does not warn about. Devon suggests they try pair programming for the last two weeks so the whole team understands the code, but Marcus worries it will slow him down.

What happens next?

The team has to decide how to finish the app and who does what. Tempers are a little short — everyone cares about the project, but they are working in different directions.

Application Questions — Model Answers

1. Explain how the team's mix of different strengths and backgrounds could make TideTrack a better app than any one of them could build alone. Use at least two specific team members as examples.

Model Response: The team's variety of talents and perspectives is exactly what should make TideTrack stronger. Priya's design skill makes the app usable; Marcus's coding experience makes it work; Lin's firsthand knowledge of the trail means the app warns about real hazards (like the flooding spots) that an outsider would miss; and Devon's organizing keeps the team coordinated. Because Lin actually walks the

trail, she brings a user's perspective the others lack — this is a clear case of diverse perspectives improving a computing innovation and helping the team avoid building something that fails its real users.

2. Explain why it is a risk that Marcus has written most of the code alone and the others do not understand it. Connect your answer to collaboration.

Model Response: When one person writes all the code alone, the team loses the benefits of collaboration: no one is reviewing Marcus's work for bugs, and the other three cannot help fix or extend it because they do not understand it. If Marcus gets sick or busy, the whole project stalls. Working alone also means the design and the trail knowledge never get fully built into the code, so the product is weaker than it would be if the team shared the work.

3. Describe how pair programming could help this team in the last two weeks. Name the two roles and explain what each person would do.

Model Response: Pair programming puts two students at one computer in two roles. The driver types the code and focuses on the current line; the navigator reviews each line, watches for mistakes, and keeps the overall plan in mind. The partners swap roles regularly. For TideTrack this means Marcus could navigate while a less experienced teammate drives, so the knowledge spreads across the team and more than one person understands the code. It also gives Lin a way to make sure the hazard warnings get coded correctly.

4. Determine which interpersonal skills the team most needs right now to settle their disagreements, and give one concrete example of how they would use one of those skills in this situation.

Model Response: The team most needs communication, consensus building, conflict resolution, and negotiation. For example, using negotiation, Marcus could agree to try pair programming for two weeks if Priya agrees to simplify the parts of her design that would be slow to build — each side gives a little. Consensus building would help them land on a final plan everyone can support, and treating the design-versus-speed disagreement as a shared problem (conflict resolution) keeps it from becoming personal.