Codesmith’s Guide To:
How to Technically Communicate - Best Practices

---

1. **Comunicate overall strategy.**
    a. When beginning a problem, the first thing you should do is communicate your overall strategy for solving it.
    b. This verbal explanation should be high level and does not have to be extremely technical or specific. It only has to give your interviewer the sense of where you are going with your solution.
    c. You can think of this as being high level verbal pseudo code.

2. **Pseudo Code**
    a. Next, you should pseudo code your solution.
    b. Pseudo code is used to keep you on track and to break down your entire solution into more manageable pieces. There is nothing worse than getting lost in your own code.
    c. You should be focusing your pseudo code on the main pieces of your solution (e.g., where loops should be, key pieces of logic, what should be returned, etc.).
    d. Your pseudo code should be slightly more technical and specific than your verbal overall strategy from the previous step.

3. **Communicate Line By Line**
    a. This should be solid and concise explanations of each line of code.
    b. You can explain and then code the line or vise versa or a combination of both. The main thing is that you should keep it consistent throughout the problem.
    c. Beyond just what you are coding, this is your opportunity to explain WHY it is that you are coding it. You can expand on ways you are using best practices, using built in methods, writing code that is more efficient.

4. **Communicate When You Don’t Know What To Do**
    a. Any good technical interview should push you to something that you don’t know.
    b. It is not a question of if you hit a block, it is a question of how you handle the block when you hit it.
    c. Don’t panic! Speak clearly and concisely. Even if you don’t know why you are getting an error you should explain your thought process and why it might be happening.

5. **Optional Summary Explanation**
    a. Once your code is working you can optionally give a final summation. This step is advisable especially if you had a difficult time solving the problem.
    b. This is your chance to give a smooth overall explanation of what you coded line by line.
    c. This tends to be less choppy than your line by line explanation since you already have the logic done.