

Tips for Solving Programming Problems

1. Getting Started:
 - a. What are the goals of the problem?
 - b. What are the inputs?
 - c. What are the outputs?
 - d. What is their relationship?
 - e. Can you solve a small example by hand?
 - f. Is there a part of the problem you could write code for? (and worry about the rest later?)
 - g. Can you describe the algorithm in words (pseudocode)?
2. Compiling Code
 - a. What line is the syntax error on?
 - b. What does the text of the error mean?
 - c. Can you simplify your code to eliminate this problem and confirm its precise location?
3. Getting Code to Function
 - a. What evidence do you have that the code doesn't work?
 - b. What test case doesn't work and what incorrect behavior or output results?
 - c. Could you come up with a simpler example that demonstrates the error?
 - d. What lines of code might be producing the bug?
 - e. Why - what hypotheses do you have for what might be causing the problem?
 - f. How can you test these hypotheses? (e.g. writing new test cases, adding print statements, using a debugger)
 - g. Could you walk through an example that doesn't work: by hand? with a debugger?

Adapted from [CSTeachingTips.org](https://www.cs-teaching-tips.org)