# How to Approach Coding Challenges

A common format of modern software developer interviews is being asked to solve a programming problem – similar to what you might see on a platform like Hackerrank or Coderbyte. Solving these types of problems is a skill that you can develop over time. Here’s an example of a prompt you might be asked to solve:

*Write a method that takes an array of strings as a parameter and returns the average length of all of its strings.*

What is the best approach to take if you don’t know where to start? Here’s a list of steps you can follow.

1. **Understand** the prompt

Before you start trying to solve the prompt with code, you need to understand what it is asking you to do. **Look for sample input and outputs**, or create your own if not given. For the prompt above, we can create this example:

*[“hello”, “foo”, “barbaz”, “goodbye”] -> length of each string: [5, 3, 6, 7]*

*The average length of the strings is: (5+3+6+7)/4 = 21/4 = 5.25*

If you still don’t understand the problem after reviewing an example, or if the prompt is ambiguous – ask clarifying questions to your interviewer. For example, “are the strings guaranteed to be in a particular order or not? Could there be empty strings?”

1. **Break the problem down** into the steps needed to solve the problem

Before you write any code, write down the steps you’ll need to complete the solution. You can do this in a comment block above your solution code. For example:

*A) Find the length of each string in the array*

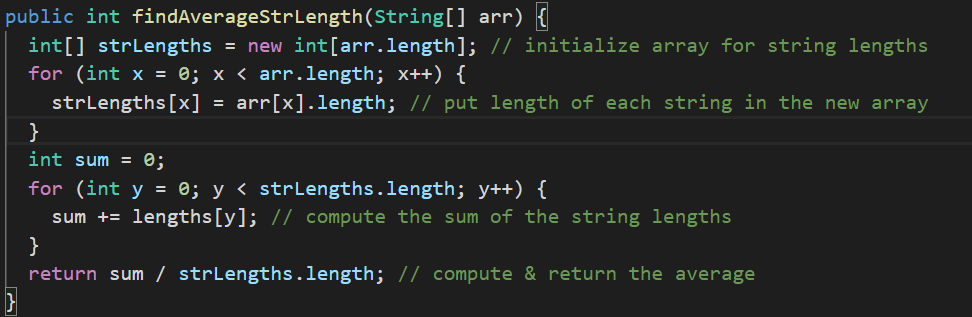
*B) Insert these new numbers into a new array*

*C) Sum the numbers in this new array*

*D) Divide the sum by the length of the array to find the answer*

1. **Write the code** to solve the problem

Translate the steps you’ve identified above into code which does that step. For example, for step (A) above you may realize you need to write a for loop to iterate through the array.



1. **Explain** your solution

Always try to **think out loud** as you are going through the previous steps, and explain your thought process as you solve the problem. This will help your interviewer follow you and let them know you are on the right track, and even if you don’t finish with a fully correct solution you may get some credit.

1. **Run & Debug** your code

Most platforms have a way for you to run your code and test whether you have passed at least one of the test cases. Use this liberally if you have the ability, because this will let you know if you have made a mistake somewhere. Use usual debugging strategies like printing out values, proofreading code, and don’t forget to read any stacktraces from exceptions that are thrown.

1. **Optimize** your code

When you first solve the problem, it is best to take the approach that is easiest to understand and write code for. However, this may not be the fastest or most efficient way to solve the problem. Often, you can find a better solution. Once you have a working solution, you can explore other ways you may have solved the problem.

For example, in our current solution, we have to complete two loops over two different arrays. We can reduce this to a single pass over the array if we compute a running sum as we iterate. We should also handle the edge case of an empty array:

