# Task 3
# Model Answer – Video 2
## Video Transcript

Hi I'm Maria Fernandez, and my pronouns are she and her.

I'm a leading scientist in BCG working from the Toronto office.

Here, I specialize in our consumer customers, which basically means personalization.

These projects contain a mix of machine learning, software engineering and a lot of AB testing.

Coding tests usually allow us to understand how you would translate a very specific problem into a code that is efficient, clear and concise.

Coding tests allow interviewers to understand your level of seniority, so let's talk about strategy to get the most out of them.

First of all, what's essential about coding tests is that the code works and it's easy to read. So with that in mind, remember to add a couple of lines of comments at the beginning that explain your solution and comments throughout the code that always helps the interviewer reading through.

And second, remember, it's very difficult to get the best solution on an interview, so if there are pieces that you think you could do better with more time, just add a comment and say it.

If you hit those three things - a code that works, a code that's well commented, and that you're finding yourself ways to improve it, that will impress any interviewer.

If we think of our specific stairs coding problem, there's a very good example on how you can solve something easily, but also how can you solve it in efficient way.

An easy way of solving it is to implement it recursively, but you'll realize the problem will branch out really, really quickly, making it not efficient in terms of computing time.

If we think about the problem for just a sec, we can realize that the memory will allow us to make it much more efficient in terms of timing, because the path you need to explore to go upon a three stair, three steps stairs is very similar for the five.

So all of the things that we learn on those three ones, you can implement that as a memory to then explore only the remaining path for that five stairs.

In terms of the biggest no no's for a coding test, two main things.

One - non organized code. It is really difficult to read a code that has dropped code in between temporarily, so make sure you clean up that before submitting.

Second - non-working code. It is very difficult to review a code that's not working. It doesn't allow us to know if it was just a simple bug that you didn't realize or if the coder didn't know how to implement it correctly.

So remember to make sure that your code runs top to bottom.

Finally, this applies to all interviews but especially important for coding.

Remember to pause and breathe once you've read through the problem and to think before you start into coding mode.

It is really difficult to get the most efficient solution if you are not thinking beforehand.

Second, remember, coding is just a fun big puzzle.

So remember to enjoy all of the interviews and the challenges.