I once had an issue with git, I simply googled the error, and checked stack overflow for a solution. Thankfully some good samaritan had already outlined steps to solve the issue alongside the command I don’t even remember what was. Changed directory, ran commands, issue solved! I gave my little thanks by upvoting the answer before going about my day.

In that case, you could simply say I did nothing but copy and paste but is that always the reality of the life of a Software Developer?

What if in this situation where you are to solve an algorithm problem in say C programming language and you just can’t find a solution? The closest solution you find to the problem is a slight variation to the algorithm problem you are trying to solve and then it is in a different programming language! How do you solve that? GOTCHA! Hooked!

That’s where understanding what you are doing comes into play!

In such situations, you need to UNDERSTAND how that piece of code works. This will enable you to refactor your code to suit your program. You won’t be able to do this if all you want to do is simply copy and paste.

Software Engineering is all about solving real-life problems using tools like programming languages and other pieces of software. There cannot always be a copy-and-paste solution to everything. Some solutions require you to come up with something genuine or better still apply the knowledge you have in your own way.

If all that Software Engineers do is simply google up solutions, and copy and paste, then they will not be paid that high for jobs that easy.

If you are consoling yourself after copying a piece of code or someone else’s entire code with “…at least programmers do a lot of copying and pasting” then I am really sorry to say, you might not end up being a good programmer.

Of course, there will be cases where you might need to copy, some problems could be a little more advanced than our current skill set, a bit confusing where you need some extra help. The fundamental thing to do at those times is to ensure you understand what you are copying at the very least. This will enable you to apply this knowledge when you see something similar again.