7 Tips To Write Clean And Better Code in 2020

**1. Use Meaningful Names**

You will be writing a lot of names for variables, functions, classes, arguments, modules, packages, directories and things like that. Make a habit to use meaningful names in your code. Whatever names you mention in your code it should fulfill three purposes…*what it does*, *why it exists* and *how it is used*. For Example:

int b; // number of users.

In the above example, you need to mention a comment along with the name declaration of a variable which is not a characteristic of a good code. The name that you specify in your code should reveal it’s intent. It should specify the purpose of a variable, function or methods. So for the above example, a better variable name would be:- *int number\_of\_users*. It may take some time to choose meaningful names but it makes your code much cleaner and easy to read for other developers as well as for yourself. Also, try to limit the names to three or four words.

### 2. Single Responsibility Principle (SRP)

Classes, Functions or methods are a good way to organize the code in any programming language so when you are writing the code you really need to take care that how to write a function that communicates it’s intent. Most of the beginners do this mistake that they write a function that can handle and do almost everything (perform multiple tasks). It makes your code more confusing for developers and creates problems when they need to fix some bugs or find some piece of code. So when you are writing a function you should remember two things to make your function clean and easy to understand…

1. They should be small.
2. They should do only one thing and they should do it well.

The above two points clearly mention that your function should follow *single responsibility principle*. Which means it shouldn’t have nested structure or it should not have more than two indent level. Following this technique make your code much more readable and other developers can easily understand or implement another feature if your function fulfills a specific task.  
Also, make sure that your function should not have more than three arguments. More arguments perform more tasks so try to keep the arguments as less as possible. Passing more than three arguments makes your code confusing, quite large and hard to debug if any problem would be there. If your function has try/catch/finally statement then make a separate function containing just the try-catch-finally statements.  
Take care of your function name as well. Use a descriptive name for your function which should clearly specify that what it does.

*Example:*

function subtract(x, y) {

return x - y;

}

In the above example the function name clearly shows that it’s purpose is to perform subtraction for two numbers, also it has only two arguments.

### 3. Avoid Writing Unnecessary Comments

It’s a common thing that developers use comments to specify the purpose of a line in their code. It’s true that comments are really helpful in explaining the code what it does but it also requires more maintenance of your code. In development code move here and there but if the comment remains at the same place then it can create a big problem. It can create confusion among developers and they get distracted as well due to useless comments. It’s not like that you shouldn’t use comments at all, sometimes it is important, for example…if you are dealing with third party API where you need to explain some behavior there you can use comments to explain your code but don’t write comments where it’s not necessary.  
Today modern programming languages syntax are English like through and that’s good enough to explain the purpose of a line in your code. To avoid comments in your code use meaningful names for variables, functions or files.

### 4. Write Readable Code For People