WRITE TO BE REVIEWED

Best Practice 4 - Code should be written to be reviewed

While writing your software code, keep in mind that someone is going to review your code and you will have to face criticism about one or more of the following points but not limited to:

- Bad coding

- Not following standard

- Not keeping performance in mind

- History, Indentation, Comments are not appropriate.

- Readability is poor

- Open files are not closed

- Allocated memory has not been released

- Too many global variables.

- Too much hard coding.

- Poor error handling.

- No modularity.

- Repeated code.

Keep all the above-mentioned points in your mind while coding and stop them before they jump in your source code. Once you are done with your coding, go for a self-review atleast once. I'm sure, a self-review would help you in removing 90% problems yourself.

Once you are completely done with your coding and self review, request your peer for a code review. I would strongly recommend to accept review comments happily and should be thankful to your code reviewers about the comments. Same time, it is never good to criticize any source code written by someone else. If you never did it, try it once and check the coder's expression.

Accept criticism but don't criticize

Poorly written source code teaches you to write good source code provided you take it positively and learn a lesson from it.

BUG FREE

Your target should be to stop the bugs at first place and create a BUG-FREE code. Think like a tester, so that you should have a challenge for the testers.